Funktionsbeschreibung Ausgabe 01/2004

motion control

information system

SINUMERIK 840D/840Di/810D
Rechnerkopplung RPC SINUMERIK

SIEMENS

SIEMENS

Motion Control Information System

SINUMERIK 840D/840Di/810D Rechnerkopplung RPC SINUMERIK

Funktionsbeschreibung

Gültig für

Steuerung	*Softwarestand*
Motion Control Information System	1

Ausgabe 01.04

SINUMERIK®-Dokumentation

Auflagenschlüssel

Die nachfolgend aufgeführten Ausgaben sind bis zur vorliegenden Ausgabe erschienen.

In der Spalte "Bemerkung" ist durch Buchstaben gekennzeichnet, welchen Status die bisher erschienenen Ausgaben besitzen.

Kennzeichnung des Status in der Spalte "Bemerkung":

A .... Neue Dokumentation.
B .... Unveränderter Nachdruck mit neuer Bestell-Nummer.
C .... Überarbeitete Version mit neuem Ausgabestand.
Hat sich der auf der Seite dargestellte technische Sachverhalt gegenüber dem vorherigen Ausgabestand geändert, wird dies durch den veränderten Ausgabestand in der Kopfzeile der jeweiligen Seite angezeigt.

Ausgabe	**Bestell-Nr.**	**Bemerkung**
01.04	6FC5297-6AD61-0AP0	**A**

Dieses Buch ist Bestandteil der Dokumentation auf CD-ROM (DOCONCD)

Ausgabe	**Bestell-Nr.**	**Bemerkung**
09.02	6FC5 298-7CA00-0AG0	**C**

Weitere Informationen finden Sie im Internet unter:
http://www.ad.siemens.de/sinumerik

Die Erstellung diese Unterlage erfolgte mit WinWord V 8.0 und Designer V 7.0 und dem DokuTool AutWinDoc .

Bestell-Nr. 6FC5297-6AD61-0AP0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft.

Vorwort

Gliederung der Dokumentation

Die SINUMERIK-Dokumentation ist in 3 Ebenen gegliedert:

- Allgemeine-Dokumentation
- Anwender-Dokumentation
- Hersteller/Service-Dokumentation

Nähere Informationen zu Druckschriften über SINUMERIK 840D/810D sowie Druckschriften, die für alle SINUMERIK-Steuerungen gelten, erhalten Sie von Ihrer SIEMENS-Niederlassung.

An wen wendet sich das Handbuch

Die vorliegende Dokumentation wendet sich an den Hersteller von Werkzeugmaschinen mit SINUMERIK 840D bzw. SINUMERIK 810D und SIMODRIVE 611D.

Hotline

Bei Fragen wenden Sie sich bitte an folgende Hotline:

A&D Technical Support
Tel.: +49 (0) 180 / 5050 - 222
Fax: +49 (0) 180 / 5050 - 223
E-Mail: ad.support@siemens.com

Bei Fragen zur Dokumentation (Anregungen, Korrekturen) senden Sie bitte ein Fax an folgende Faxadresse oder email:
Fax: +49 (0) 9131 / 98 -2176
E-Mail: motioncontrol.docu@erlf.siemens.de

Faxformular: siehe Rückmeldeblatt am Schluss der Druckschrift.

Internetadresse SINUMERIK

http://www.ad.siemens.de/sinumerik

Gegenstand des Handbuches

Die Beschreibung erklärt die Rechnerkopplungssoftware auf der PCU-Komponente (Liefergegenstand) und die erforderliche Funktionalität, die eine zu erstellende Fertigungsleitrechner-Software kommunikationsseitig erbringen muss. Die nötigen Schnittstellen sind in der Beschreibung enthalten.

Die Druckschrift besteht aus:
Teil 1: Nahtstelle zum Fertigungsleitrechner
Teil 2: Nahtstelle zu PLC/NCK.

Suchhilfen

Zu Ihre besseren Orientierung wird Ihnen ein Inhaltsverzeichnis angeboten sowie die Anhänge:

- Literaturverzeichnis
- Index

Gefahr- und Warnkonzept

In dieser Druckschrift werden folgende Gefahr- und Warnhinweise verwendet. Hier die Symbolerläuterung:

Gefahr

Dieser Warnhinweis bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Warnung

Dieser Warnhinweis bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

Dieser Warnhinweis (mit Warndreieck) bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

Dieser Warnhinweis (ohne Warndreieck) bedeutet, dass ein Sachschaden eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Achtung

Dieser Warnhinweis bedeutet, dass ein unerwünschtes Ereignis oder ein unerwünschter Zustand eintreten **kann**, wenn der entsprechende Hinweis nicht beachtet wird.

Weitere Hinweise

!

Wichtig

Dieser Hinweis bedeutet, dass ein wichtiger Sachverhalt zu beachten ist.

Hinweis

Dieses Symbol erscheint in dieser Dokumentation immer dann, wenn weiterführende Sachverhalte angegeben werden.

Maschinenhersteller

Das abgebildete Symbol erscheint in dieser Dokumentation immer dann, wenn der Maschinenhersteller das beschriebene Funktionsverhalten beeinflussen oder verändern kann. Beachten Sie die Angaben des Maschinenherstellers!

Warenzeichen

IBM® ist eingetragenes Warenzeichen der International Business Corporation. MS-DOS® und WINDOWS™ sind eingetragene Warenzeichen der Microsoft Corporatiion.

■

Motion Control Information System

SINUMERIK 840D/840Di/810D
Rechnerkopplung RPC SINUMERIK

Nahtstelle zum Fertigungsleitrechner (NFL) (FBR-NFL)

1

1 Einführung

Beschreibung der Schnittstelle zwischen einem Fertigungsleitrechner (FLR) und einer SINUMERIK 840D oder 810D.

- Die Kommunikation zwischen den Systemen basiert auf Ethernet und TCP/IP-Protokoll.
- Die Initiative zur Datenübertragung kann sowohl vom FLR als auch vom SINUMERIK ausgehen.
- Die Schnittstelle zu RPC SINUMERIK basiert auf RPC (Remote Procedure Call) und der Übertragung von Dateien nach DCE-Standard.

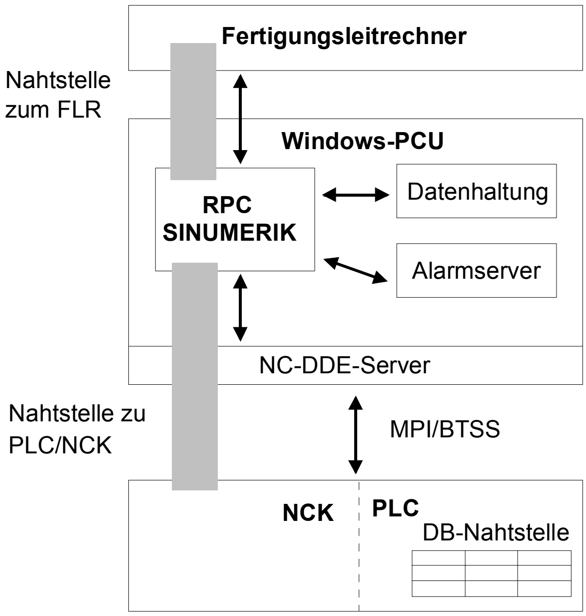


Bild 1-1 Schnittstellen

Datenübertragung

Bei wenigen zu übertragenden Daten werden diese im RPC-Funktionsaufruf übergeben. NC-Programme und andere Daten, die bereits als Dateien vorliegen, werden auch als Dateien übertragen. Umfangreichere Daten wie z.B. Werkzeugdaten werden in eine Datei geschrieben und dann ebenfalls als Datei übertragen.
Der Dateiname der Datei sowie weitere Parameter werden mit einem RPC-Funktionsaufruf dem Koppelpartner übergeben.
Abhängig von den Übertragungsmöglichkeiten zwischen FLR und SINUMERIK werden die Dateien entweder mit FTP übertragen, oder bei einem Netzwerk-Filesystem, durch kopieren.

RPC-Funktionen

Die RPC-Funktionen in dieser Dokumentation sind aus Sicht des Fertigungsleitrechners und damit des Anwenders beschrieben. Am Fertigungsleitrechner ankommende RPC-Calls werden nachfolgend als „Aufgerufene Funktion“ bezeichnet. Ein im Fertigungsleitrechner ankommender RPC-Call bewirkt, dass die entsprechende Funktion im FLR angesprungen wird. Vom FLR aufzurufende RPC-Calls werden nachfolgend als „Funktionsaufruf“ bezeichnet. Der Funktionsaufruf des FLR bewirkt, dass in RPC SINUMERIK die entsprechende Funktion aufgerufen wird.

COM-Schnittstelle

Das RPC SINUMERIK-OCX Entwicklungspaket stellt ein Zusatzprodukt zu der RPC SINUMERIK Rechnerkopplung dar. Die MCIS_RPC.OCX kapselt die RPC–Schnittstelle von RPC SINUMERIK in COM-Aufrufe (Component Object Model). Unter Verwendung der MCIS_RPC.OCX-Schnittstelle wird die RPC SINUMERIK–Schnittstelle für eine ganze Reihe von verbreiteten Windows–Entwicklungssystemen zugänglich, ohne dass eine C/C++ Programmierung notwendig wird.

Die RPC SINUMERIK-OCX-Beschreibung finden Sie in **Kapitel 10** des ersten Teiles dieser Druckschrift (NFL).

■

2

2 Systeminstallation

2.1 Systemanforderungen

2.1.1 Software

SINUMERIK 840D/810D

Die folgende Software wird auf der SINUMERIK 840D/810D benötigt:

- PCU50/70 V1/V2, HMI-ADVANCED 6.x
- Treiber für Ethernet-Anschaltung und TCP/IP müssen enthalten/installiert sein.
- MCIS-TDI (Modul Toolhandling) (nur wenn die Funktionen x_DATA_x SFkt 21-23 genutzt werden sollen).
- RPC SINUMERIK Rechnerkopplungs-SW-Paket

Fertigungsleitrechner

Treiber für die installierte Ethernet-Anschaltungsbaugruppe.
TCP/IP mit den Diensten FTP oder NFS und RPC nach DCE-Standard.

2.1.2 Hardware

SINUMERIK 840D/810D

- PCU50
- Eingesetzte Netzwerkanschaltungen können bei der SINUMERIK Hotline erfragt werden oder siehe SINUMERIK FAQ's: (http://www.ad.siemens.de/support)

Fertigungsleitrechner

Eingesetzte Netzwerkanschaltungen können bei der SINUMERIK Hotline erfragt werden oder siehe SINUMERIK FAQ's: (http://www.ad.siemens.de/support)

Verbindung SINUMERIK-FLR

Für die Ethernet-Verbindung wird ein Netzwerkkabel verwendet.

2.2 Inbetriebnahme

Die RPC SINUMERIK Software wird auf die SINUMERIK aufgespielt und die Ini-Datei des Programms konfiguriert.
Auf der PLC muss der DB12 für die DB-Nahtstelle eingerichtet werden. Dies wird voraussichtlich zusammen mit der Installation der PLC-Software des Maschinenherstellers erfolgen.
Der Aufbau des Datenbausteins ist in „Rechnerkopplung RPC SINUMERIK, Nahtstelle zu PLC/NCK (NPL)“ in **Kapitel 1** (Siehe /2/ Kap. 1 Nahtstelle RPC und Maschinen-PLC) beschrieben.

■

Für Notizen

3

3 Betriebsarten und Zustände

3.1 Betriebsarten

Die SINUMERIK 840D verfügt über die Betriebsarten Automatik, MDA, JOG und TEACH IN. Die Rechnerkopplung benötigt zusätzlich ihre eigenen Betriebsarten. Das Funktionsverhalten von RPC SINUMERIK, auch im Zusammenspiel mit der PLC, ist abhängig von der Rechnerkopplungsbetriebsart. Zur Bedienung und Anzeige der Rechnerkopplungsbetriebsart wird ein eigener Dialog auf der SINUMERIK angeboten. Die Rechnerkopplungsbetriebsarten sind:

- FLR-Modus unbemannt
- FLR-Modus bemannt
- Manueller Modus
- Sondermodus
- Offline

3.1.1 FLR-Modus unbemannt/bemannt

In den beiden Rechnerkopplungsbetriebsarten FLR-Modus unbemannt/bemannt dürfen, in Verbindung mit der Betriebsart Automatik, vom FLR vorgegebene NC-Programme von der PLC gestartet werden. Die Unterscheidung in unbemannt und bemannt dient dazu, um bei Störungen unterschiedlich reagieren zu können.
Tritt bei bemannter Produktion eine Störung auf, bleibt der Werkstückträger in der Maschine. Der Bediener muss die Störung beseitigen, um die Bearbeitung fortsetzen zu können.
Bei unbemannter Produktion kann im Falle einer Störung die Bearbeitung des Werkstückträger beendet werden, um danach mit einem anderen Werkstückträger weiterzumachen.

3.1.2 Manueller Modus

Die NC-Programme werden vom FLR vorgegeben. Die Anwahl des NC-Programmes erfolgt durch die Rechnerkopplung, das angwählte NC-Programm wird in der Kopfzeile der Bedienoberfläche (BOF) angezeigt.
Der Programmstart muss durch den Bediener erfolgen.

Hinweis

In den Rechnerkopplungsbetriebsarten FLR-Modus unbemannt/bemannt und dem manuellen Modus unterliegt die Maschine dem automatischen Materialfluß, d.h Werkstückträger werden automatisch durch das Transportsystem angeliefert und nach ihrer Bearbeitung wieder abgeholt.

3.1.3 Sondermodus

Im Sondermodus werden keine automatischen Transporte zur Maschine ausgeführt, der FLR überträgt keine Programmzuordnung und es erfolgt kein automatischer Start von NC-Programmen. Die Maschine unterliegt der Kontrolle des Bedieners, die Kommunikation mit dem FLR erfolgt ansonsten jedoch in vollem Umfang. Der Sondermodus dient in der Regel zum Austesten von NC-Programmen oder zur manuellen Fertigung ungeplanter Werkstücke.

3.1.4 Offline

Offline kennzeichnet die Verbindungsunterbrechung zwischen FLR und SINUMERIK, es findet keine Kommunikation zwischen FLR und SINUMERIK statt. Die SINUMERIK setzt Offline, wenn er eine Verbindungsunterbrechung zum FLR erkennt. Ebenso kennzeichnet der FLR eine Maschine in seinen Zustandsdaten und im Anlagenabbild als Offline, sobald er eine Verbindungsunterbrechung erkennt.
Nach Offline muss am FLR eine Synchronisation der Maschine erfolgen.
Wird an der Maschine eine Betriebsart (sowohl SINUMERIK- als auch Rechnerkopplungs-Betriebsart) umgeschaltet, muss dies dem FLR mit dem RPC-Call R_MACHINE_H() gemeldet werden.

3.2 Maschinenzustände

Die Maschine kann sich in folgenden Zuständen befinden:

- Neuanlauf: nach Neustart der SINUMERIK
- untätig: es läuft keine Bearbeitung
- tätig: Bearbeitung läuft
- gestört: Bearbeitung unterbrochen
- Komponenten abgeschaltet

3.3 Haltestellen/Liegeplatzzustände

Der FLR muss die Zustände der Haltestellen kennen, um Transportaufträge an das Transportsystem stellen zu können.
Folgende Zustände sind möglich:

- freigegeben
- für TPS gesperrt
- gestört

3.4 Werkstückträgerzustände

Für die Ermittlung von Transportaufträgen benötigt der Leitrechner die Werkstückträgerzustände. Der Werkstückträger kann sich in folgenden Zuständen befinden:

- unbearbeitet - ohne Programmzuordnung:
  es wurde noch kein NC-Programm für die Bearbeitung zugeordnet, d.h. es ist noch keine Bearbeitung möglich.
- unbearbeitet - Programmzuordnung erfolgt:
- in Bearbeitung
- fertig bearbeitet
- bearbeitet mit Fehler
- nur zur Pufferung:
  der Werkstückträger wird nur auf der Maschine gepuffert, es erfolgt keine Bearbeitung.

■

4

4 Werkzeugdaten

Werkzeugdaten

Da nicht immer alle Daten eines Werkzeuges benötigt werden, gibt es 3 Varianten.

1. Die erste Variante enthält den vollen Umfang der Werkzeugdaten, die beiden anderen jeweils nur einen Teil der Daten.
2. Die Projektierbarkeit des Datenumfanges pro Werkzeug umfasst nur ganze Bereiche, d.h. für jede Variante gibt es eine Liste (in der Registry), in der die Bereiche aufgeführt sind, die bei dieser Variante übertragen werden.

Das Dateiformat für die Werkzeugdaten entspricht dem Format der Datensicherung der NC840D (Lochstreifen-/ASCII-Format lt. /BA/ ; z.B. wie in _N_TOx_TOA oder _N_TOx_INI). Die vollständige Beschreibung zu Dateninhalt und Anordnung steht in /NPL/ ab **Kapitel 4**.

Hinweis

/NPL/ steht für den zweiten Teil (NPL) dieser Unterlage.

Werkzeugdatenbereiche

Die WZ-Daten sind im NCK in verschiedenen Bereichen abgelegt.
Die Bereiche werden wie folgt bezeichnet:

Tabelle 4-1 Werkzeugdatenbereiche

Bereichsbezeichnung	**Werkzeugbezeichner**
Werkzeugdaten allgemein	$TC_TPx[y]
anwenderbezogenen Werkzeugdaten	$TC_TPCx[y]
Werkzeug-Schneidendaten	$TC_DPx[y,z]
anwenderbezogene WZ-Schneidendaten	$TC_DPCx[y,z]
Werkzeugüberwachungsdaten	$TC_MOPx[y,z]
anwenderbezogene WZ-Überwachungsdaten	$TC_MOPCx[y,z]

x : pro Bereich ein fortlaufender Wert, um einen eindeutigen Namen für die Systemvariable zu erhalten.
Y : T-Nummer
z : Schneidennummer

Tabelle 4-2 Magazinplatzdaten

Bereichsbezeichnung	**Werkzeugbezeichner**
Magazin und Platznummer	$TC_MPP6[y,z]=x

x : T-Nummer
Y : Magazinnummer
z : PLatznummer.

Aufbau der Datei

Tabelle 4-3 Beschreibung der Werkzeug- und Magazinplatzdaten

	Beschreibung
$TC_TP1[3]= 2	Duplonummer
$TC_TP2[3]= “4711“	Identnummer
$TC_TP3[3]=1	Größe nach links in Halbplätzen
$TC_TP4[3]=1	Größe nach rechts in Halbplätzen
$TC_TP5[3]=1	Größe nach oben in Halbplätzen
$TC_TP6[3]=1	Größe nach unten in Halbplätzen
$TC_TP7[3]=2	Magazinplatztyp
$TC_TP8[3]=131	Status
$TC_TP9[3]=0	Art der Werkzeugüberwachung
$TC_TP10[3]=2	Ersatzwechsel-Strategie
$TC_TP11[3]=0	Werkzeuginformation
$TC_DP1[3,1]=0	Schneidenparameter 1
$TC_DP2[3,1]=0	Schneidenparameter 2
$TC_DP3[3,1]=0	Schneidenparameter 3
$TC_DP ...	...
$TC_DP24[3,1]=0	Schneidenparameter 24
$TC_DP25[3,1]=0	Schneidenparameter 25
*$TC_MPP6[1,5]=3	T-Nummer

*
$TC_MPP6 zeigt, welches Werkzeug (T-Nummer) im angegebenen Magazin und Platz liegt.
In diesem Fall liegt das Werkzeug mit der T-Nummer 3 im Magazin 1 auf Platz 5.

■

5

5 Kommunikation zwischen FLR und SINUMERIK

5.1 Einführung Remote Procedure Calls

Die Kommunikation zwischen Fertigungsleitrechner und SINUMERIK benutzt das Konzept der Remote Procedure Calls (RPC) für die Übertragung kleinerer Datenmengen. Bei der Kommunikation über RPC wird der Kommunikationspartner beauftragt, eine mit Prozedurnamen identifizierte Funktion mit den im Aufruf enthaltenen Parametern (Daten) auszuführen.

Hinweis

Die Interface Definition Language IDL (programmiertechnische Festlegung) der im Folgenden verwendeten Aufrufe finden Sie im Anhang

---

Für die in der Folge beschriebenen Prozeduren sind die Namen nach folgendem Konzept strukturiert:

5.1.1 Struktur der Prozedurnamen

Die Prozedurnamen enthalten drei Komponenten:
1. Kommandokennung (erstes Zeichen)
2. Daten/Funktionsbezeichner
3. Empfängerkennung (letztes Zeichen)

Kommandokennung

Die Kommandokennung erscheint im ersten Namenszeichen:

C	Befehlsaufruf	(Command)
R	Aufforderung Daten zu empfangen	(Receive)
T	Aufforderung Daten zu senden	(Transmit)

Beispiel: **T_**MACHINE_M ().

Daten/ Funktionsbezeichner

Der Bezeichner kennzeichnet die Art der Daten, die angefordert oder übertragen werden oder die Funktion, für die die Daten bestimmt sind.
Beispiele: T_**MACHINE_**M (), R_**NC4WPC_**M().

Empfängerkennung

Die Empfängerkennung kennzeichnet den Adressat, der die Funktion bearbeiten soll:

H	Empfänger ist der Fertigungsleitrechner FLR	(Host)
M	Empfänger ist die Maschine	(Maschine)

5.1.2 Generell benutzte Parameter

Host

Bezeichner für den Fertigungsleitrechner mit max. 16 Zeichen. Bei Vernetzung mehrerer Maschinen mit mehreren Fertigungsleitrechnern identifiziert Host eindeutig den Fertigungsleitrechner, mit dem Daten ausgetauscht werden sollen.

Maschine

Bezeichner für eine Maschine (Max. 16 Zeichen). Im Netz müssen alle existierenden Maschinen eindeutig identifizierbar sein.

OrderNum

Auftragsnummer: Die Nummer kann optional benutzt werden, wenn RPC-Anforderungen und deren Antworten einander eindeutig zugeordnet werden müssen.

Hinweis

Bei Parametern vom Typ String muss die Stringbegrenzung durch ‘\0’ erfolgen.
In Visual C++ bedeutet "\0" und "" einen Leerstring; in Visual Basic ergibt "" einen Leerstring.
Die zulässigen maximalen Stringlängen sind bei den einzelnen Parametern genannt.

5.2 Kommunikationsabläufe

Voraussetzungen

Für die fehlerfreie Kommunikation zwischen FLR und einer oder mehreren Maschinen ist es erforderlich, dass der FLR die Kommunikationspartner kennt, von denen er RPCs bearbeiten soll. Die Maschinennamen dieser Maschinen (Clients) müssen im FLR hinterlegt sein.

Parameter

Parameter sind als Bestandteil des RPC mit übertragene Daten, über deren Bedeutung für FLR und Maschine der bei jedem RPC mit beschriebene Tabellenblock Auskunft gibt.

Quittung

Wie bei lokalen Prozeduren erfolgt bei RPCs eine positive oder negative Quittierung über den Returnwert. Diese Quittung kann bei Anforderungen, deren Verarbeitung asynchron erfolgt, nur den Empfang der Anforderung quittieren. Nach der Verarbeitung, oder wenn ein Fehler bei der Verarbeitung auftritt, muss der Kommunikationspartner mit einer entsprechenden RPC-Meldung verständigt werden. Eine Anzeige von Fehlermeldungen an der SINUMERIK erfolgt mit Hilfe des Alarmservers. Bei fehlerfrei ausgeführten Aufrufen ist der Returnwert = 0. Eine Auflistung der Returnwerte im Fehlerfall ist im Anhang enthalten.

Hinweis

Die FLR Software muss die aufgerufenen Funktionen möglichst schnell returnieren, weil während eines RPC's der Teil von RPC SINUMERIK, der den RPC absetzt, in diesem Aufruf wartet und also keine weitere Verarbeitung erfolgt. Innerhalb der im FLR aufgerufenen Funktion sollten die im RPC enthaltenen Daten in einen Pufferbereich umkopiert und danach die Funktion returniert werden. Die eigentliche Verarbeitung im FLR sollte erst danach stattfinden.

5.3 Maschinenzustandsdaten

5.3.1 Maschinenzustandsdaten senden

Aufgerufene Funktion

R_MACHINE_H (Host,
Machine,
OrderNum,
MachineMode,
MachineStatus,
NCProgramm,
ClampCubeSide,
DockPos,
DockPosStatus,
WPC,
WPCStatus,
Resint1,
Resint2,
Resbyte**)**

Übertragungsrichtung: **SINUMERIK → FLR**

Bedeutung

Maschinenzustandsdaten zum Fertigungsleitrechner schicken.

Daten

Tabelle 5-1 Parameter der Maschinenzustandsdaten-Übertragung

Parameter	Beschreibung	Format
Host	Name des Fertigungsleitrechners (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
MachineMode	Betriebsart • SINUMERIK-Betriebsart: 1: Automatik 2: MDA 4: JOG 8: TEACH IN • Rechnerkopplungs-Betriebsart: 100: FLR-Modus unbemannt 200: FLR-Modus bemannt 300: Manueller-Modus 400: Sondermodus 500: bei Bit 6=1 im RPC SINUMERIK-Modus 600: bei Bit 7=1 im RPC SINUMERIK-Modus	4 Byte (long int)
MachineStatus	Maschinenzustand 0: Neuanlauf 1: untätig 2: tätig 3: gestört 4: Komponenten abgeschaltet	4 Byte (long int)
NCProgramm	aktuelles NC-Programm*	128 Byte string
ClampCubeSide	Seite bei Spannwürfel**	4 Byte (long int)
DockPos[3]	Haltestellennummer Die Haltestellennummer entspricht dem Index in der Haltestellenliste des Nahtstellen DBs beginnend bei 1. Haltestellennummer = 0 bedeutet: ‘nicht belegt’ (Siehe /NPL/ Kap. 1.1 Beschreibung)	3 x 4 Byte (long int)
DockPosStatus[3]	Haltestellenzustand 0: freigegeben 1: für TPS gesperrt 2: gestört	3 x 4 Byte (long int)
WPC[3]	Werkstückträgerbezeichnung	3 x 6 Byte (string)
WPCStatus[3]	Werkstückträgerzustand 1: unbearbeitet, ohne Programmzuordnung 2: unbearbeitet, Programmzuordnung erfolgt 4: Programmanwahl vorbereiten 8: Programmanwahl erreicht 16: in Bearbeitung 32: fertig bearbeitet 64: bearbeitet mit Fehler	3 x 4 Byte (long int)

Parameter	Beschreibung	Format
	128: nur zur Pufferung	
Resint1***	Reserve 1	4 Byte (long int)
Resint2***	Reserve 2	4 Byte (long int)
Resbyte	Reserve 3	8 Byte (string)

* NC-Programmbezeichner des gerade laufenden NC-Programmes

** Seite (bei Spannwüfel) die gerade bearbeitet wird

*** Resint1 und 2 sind an der DB-Nahtstelle der PLC aufgelegt, werden von der PLC dort Werte eingetragen, werden sie an den FLR übertragen. Diese Werte haben für die Rechnerkopplung keine Bedeutung, sie werden nur an den FLR übergeben.

Anwendungshinweise

- Die SINUMERIK muss diesen RPC bei jeder Zustandsänderung an der Maschine absetzen. Die RPC SINUMERIK Rechnerkopplungssoftware ermittelt dazu die aktuellen Daten und setzt den Call ab.
- Ausgelöst wird der Vorgang von der PLC durch Setzen eines bestimmtes Bits an der DB-Nahtstelle.
- Der FLR kann durch den Befehl **T_MACHINE_M** (Anfordern Maschinenzustandsdaten, siehe unten) ebenfalls diesen Vorgang auslösen.

Hinweis

- Sind mehr als 3 Haltestellenpositionen zu beschreiben, so muss eine eigene Maschinenbelegungsdatei übertragen werden. Siehe **Kapitel 5.12**
- Um in einer Variablen (MachineMode) die Betriebsarten beider Teilnehmer melden zu können (SIN und FLR), kann die Summe als ein Wert übertragen werden (z.B. 201 : FLR-Modus bemannt =200 und SIN-Automatik =1).
- Die RPC SINUMERIK Rechnerkopplungssoftware führt keine Plausibilitätsprüfung auf die gemeldeten Betriebsarten durch.

5.3.2 Maschinenzustandsdaten anfordern

Funktionsaufruf

T_MACHINE_M (Host,
Machine,
OrderNum)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Maschinenzustandsdaten anfordern

Tabelle 5-2

Parameter	Beschreibung	Format
Host	Name des Fertigungsleitrechners (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)

Anwendungshinweis

Der FLR kann mit diesem Call die Maschinenzustandsdaten anfordern, z.B. bei der Synchronisation. Die SINUMERIK setzt dann die Daten mit dem Befehl **R_MACHINE_H()** an den FLR ab.

Beispiel

T_MACHINE_M(“FLR1“,“BAZ3“,0);

5.4 Fertigungsdialog

Beschreibung

Nach Ankunft eines Werkstückträgers an der Maschine, sendet die SINUMERIK, auf Anstoß der PLC, die Maschinenzustandsdaten an den FLR.
Anhand der Daten erkennt der FLR, welcher Werkstückträger an der Maschine angekommen ist und überträgt die Programmzuordnung für diesen Werkstückträger.
Bei Werkstückträgern mit Spannwürfel wird für jede Seite ein NC-Programm zugeordnet. RPC SINUMERIK speichert diese Programmzuordnungen. Jede Programmzuordnung besteht aus Werkstückträger, Seite und NC-Programm. Es wird jeweils das nächste NC-Programm übertragen und angewählt. Danach darf von der PLC das NC-Programm gestartet werden (bei FLR-Modus bemannt und unbemannt). Die Maschine meldet den NC-Start und später das NC-Ende durch die Maschinenzustandsdaten.

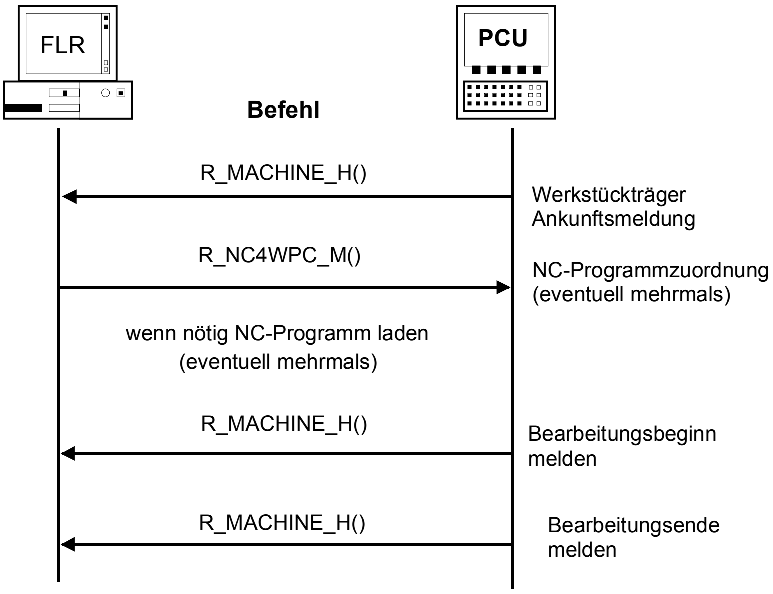


Bild 5-1 Fertigungsdialog, Normalablauf fehlerfrei

Treten während dieses Ablaufes Fehler auf, werden entsprechende Meldungen (siehe 5.5.1 Meldung von SINUMERIK an FLR) abgesetzt.

5.4.1 Programmzuordnung

Funktionsaufruf

R_NC4WPC_M (Host,
Machine,
OrderNum,
WPC,
NCProg,
Date,
NCPLength,
ClampCubeSide,
TpFlag,
NCExtern,
Resint1,
Resint2,
Resbyte)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Anweisung an die Maschine geben, welches Programm aktiviert werden soll

Daten

Tabelle 5-3 Parameter der Programmzuordnung

<table>
<tr><th>Parameter</th><th>Beschreibung</th><th>Format</th></tr>
<tr><td>Host</td><td>Name des Fertigungsleitrechners(Host)</td><td>16 Byte (string)</td></tr>
<tr><td>Machine</td><td>Maschinenname</td><td>16 Byte (string)</td></tr>
<tr><td>OrderNum</td><td>Auftragsnummer</td><td>4 Byte (long int)</td></tr>
<tr><td>WPC</td><td>Werkstückträgerbezeichnung</td><td>6 Byte (string)</td></tr>
<tr><td>NCProg</td><td>NC-Programm
NC-Programm in der Form:
“\mpf.dir\Zylinderkopf.mpf“</td><td>128 Byte (string)</td></tr>
<tr><td>Date</td><td>Datum der letzten Änderung des NC-Programmes
(Unix-Uhrzeit)</td><td>4 Byte (long int)</td></tr>
<tr><td>NCPLength</td><td>NC-Programm-Größe in Bytes</td><td>4 Byte (long int)</td></tr>
<tr><td>ClampCubeSide</td><td>Seite bei Spannwürfel</td><td>4 Byte (long int)</td></tr>
<tr><td>TpFlag</td><td>Transportkennung
= 0: kein Folgearbeitsgang
= 1: Folgearbeitsgang
= 9: nur zur Pufferung</td><td>4 Byte (long int)</td></tr>
<tr><td>NCExtern</td><td>Abarbeiten von Extern
<table>
<tr><th>Bit 1</th><th>Bit 0</th><th>Funktion</th></tr>
<tr><td>0</td><td>0</td><td>NC-Programm auf NCK abarbeiten</td></tr>
<tr><td>0</td><td>1</td><td>NC-Programm von Extern abarbeiten (dezimal 1)</td></tr>
</table></td><td>4 Byte (long int)</td></tr>
</table>

Parameter	Beschreibung			Format
	1	0	Werkstück auf NCK abarbeiten (dezimal 2)	
	1	1	Werkstück von Extern abarbeiten (dezimal 3)	
Resint1	Reserve 1			4 Byte (long int)
Resint2	Reserve 2			4 Byte (long int)
Resbyte	Reserve 3			8 Byte (string)

Anwendungshinweise

- Dieser RPC darf für einen Werkstückträger mehrfach kommen, wenn bei einem Spannwürfel jede Seite mit einem eigenen NC-Programm bearbeitet wird.
  Die Bearbeitung der Seiten findet in der Reihenfolge statt, wie die Programmzuordnungen vom FLR an RPC SINUMERIK gemeldet werden.
- Bei den Programmzuordnungen eines Werkstückträgers muss bei allen, außer der letzten, die Transportkennung "1 = Folgearbeitsgang" gesetzt sein. Ist die Folgearbeitsgangkennung für eine Seite gesetzt, verbleibt der Werkstückträger nach Ende des NC-Programmes auf dem Bearbeitungsplatz. Da bei der letzten Seite diese Kennung nicht gesetzt ist, kann man daran erkennen, dass keine weitere Bearbeitung mehr folgt, also der Werkstückträger vom Bearbeitungsplatz ausgeschoben werden kann.
- Wird ein Werkstückträger nur zur Pufferung auf der Maschine abgelegt, kann dies durch Transportkennung "9 = nur zur Pufferung" gemeldet werden. In diesem Fall wird kein NC-Programm angegeben.

Beispiel

R_NC4WPC_M ("FLR1", "BAZ3", 0, "WPC05", "\mpf.dir\Kw15.mpf", 862826400, 3210, 1, 0, 0, 0, 0, "\0");

5.5 Meldungen

5.5.1 Meldung von SINUMERIK an FLR

Aufgerufene Funktion

R_REPORT_H (Host,
Machine,
OrderNum,
Typ,
Number,
Time,
Flag,
Resint1,
Resint2,
Resbyte)

Übertragungsrichtung: **SINUMERIK→ FLR**

Bedeutung

Meldung an den Fertigungsleitrechner abgeben

Daten

Tabelle 5-4 Parameter der Meldung von SINUMERIK an FLR

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
Typ	Meldungstyp 1: Alarme 2: Bedienerunterbrechung 3: Betriebsmeldungen 4: Fehlermeldung der RPC SINUMERIK Rechnerkopplungssoftware 5: Positive Quittung bei R_DATA_M (), R_DDEDATA_M () und R_VAR_M () 6: Positive Quittung bei C_ORDER_M ()	4 Byte (long int.)
Number[10]	Meldungsnummernfeld • Feld für bis zu 10 Alarme. • Nicht benötigte Felder sind mit 0 zu belegen • Bei den anderen Meldungstypen ist immer nur Number[0] belegt	10 x 4 Byte (long int.)
Time[10]	Zeitstempelfeld • Feld für bis zu 10 Einträge. • Nicht benötigte Felder sind mit 0 zu belegen	10 x 4 Byte (long int.)
Flag[10]	kommt/geht Kennung, Feld bis zu 10 Einträge	10 x 1 Byte

Parameter	Beschreibung	Format
	C: Meldung kommt, kein Maschinenstillstand S: Meldung kommt, Maschinenstillstand G: Meldung geht L: alle Meldungen sind gegangen Bei kommenden Meldungen muss unterschieden werden: führt zu Maschinenstillstand ja/nein	
Resint1	Reserve 1	4 Byte (long int.)
Resint2	Reserve 2	4 Byte (long int.)
Resbyte	Reserve 3	8 Byte (string)

Bei R_DATA_M (), R_DDEDATA_M (), R_VAR_H () und C_ORDER_M () kann die Verarbeitung nicht synchron während des Calls erfolgen, deshalb muss nach der Verarbeitung durch RPC SINUMERIK eine positive Quittung an den FLR gesendet werden, um den FLR über den Abschluss des Vorganges zu informieren. Bei einer positiven Quittung auf R_DATA_M wird als "Fehlernummer" die SFkt von R_DATA_M benutzt, sie dient zur Zuordnung der Quittung. Bei einer positiven Quittung auf C_ORDER_M () wird ebenfalls die SFkt als Fehlernummer eingetragen, der Typ ist jedoch 6. Bei R_DDEDATA_M wird als "Fehlernummer" 1000 gemeldet und bei R_VAR_M "Fehlernummer" 0.

Hinweis

Number, Time und Flag mit gleichem Index gehören zusammen

Anwendungshinweise

Der RPC R_REPORT_H() wird benutzt, um einzelne Meldungen oder bis zu 10 Alarmmeldungen gemeinsam an den Leitrechner zu übertragen.
Sonderfall: Ist auf der Maschine der letzte Alarm gegangen, so wird dieser Zustand mit R_REPORT_H() und den Parametern:

- Typ = 1
- Number[0] = 0
- Flag[0] = L

an den Leitrechner gesendet.

Meldungsfilter

In RPC SINUMERIK wurde ein Meldungsfilter eingebaut mit dessen Hilfe die Weiterleitung bestimmter Meldungen oder Alarme an den FLR unterdrückt werden kann. Es gibt die Möglichkeit Alarmtypen des Alarmservers komplett zu unterdrücken und auch die Möglichkeit Nummernbereiche zu unterdrücken.
Die Projektierung erfolgt mit der Datei ScAlarmEx.ini im ADD_ON Verzeichnis.
Die Sektion [AlarmTypeExclusion] dient zu Ausschluss von Alarmtypen, die Sektion [AlarmRangeExclusion] zum Ausschluss von Nummernbereichen.
Die Schlüsselworte (Type1 bis TypeN bzw. Range1 bis RangeN) müssen lückenlos vergeben werden, die Lesefunktion liest mit aufsteigender Nummer und hört bei der ersten nicht vorhandenen Nummer auf.

Der Aufbau von ScAlarmEx.ini:

```
[AlarmTypeExclusion]
Type1=6
Type2=7
Type3=8
Type4=9
Type5=10
Type6=11
[AlarmRangeExclusion]
Range1=100,199
Range2=250,250
Range3=2000,3999
Range4=5000,5050
Range5=6799,6799
```

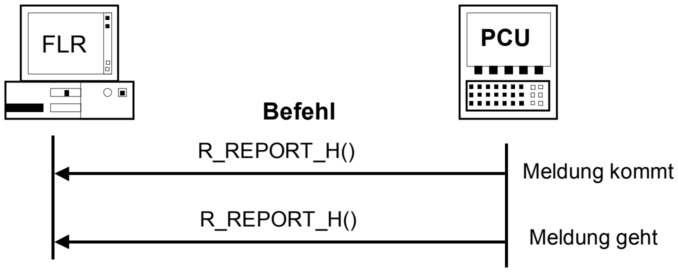


Bild 5-2 Ablauf für Alarme/ Bedienerunterbrechungen/ Betriebsmeldungen

Wird die Bearbeitung an der Maschine nach Beendigung von Alarmen, oder Bedienerunterbrechungen mit Maschinenstillstand fortgesetzt, muss dies mit dem RPC **R_MACHINE_H ()** mit dem Maschinenzustand dem FLR gemeldet werden (beim Transportsystem entsprechend mit **R_TPS_H ()** .
Eine Auflistung der Fehlermeldungen für Meldungstyp 4 ist im Anhang enthalten.

5.5.2 Meldung von FLR an SINUMERIK

Funktionsaufruf

R_REPORT_M (Host,
Machine,
OrderNum,
Typ,
Number,
Resint1,
Resint2,
Resbyte

Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Fehlermeldungen des FLR werden an RPC SINUMERIK in der SINUMERIK gesendet.

Daten

Tabelle 5-5 Parameter der Meldung von FLR an SINUMERIK

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
Typ	Meldungstyp 4: Fehlermeldungen des FLR an SINUMERIK	4 Byte (long int.)
Number	Fehlernummer	4 Byte (long int.)
Resint1	Reserve 1	4 Byte (long int.)
Resint2	Reserve 2	4 Byte (long int.)
Resbyte	Reserve 3	8 Byte (string)

Anwendungshinweise

Die Fehlermeldungen werden auf der SINUMERIK in ein LOG-File eingetragen. Zusätzlich lösen bestimmte Fehlermeldungen eine Fehlerbehandlung in RPC SINUMERIK aus.

Beispiel

R_REPORT_M ("FLR1", "BAZ3", 0, 4, -13, 0, 0, "\0");

5.5.3 Anstehende Alarme anfordern

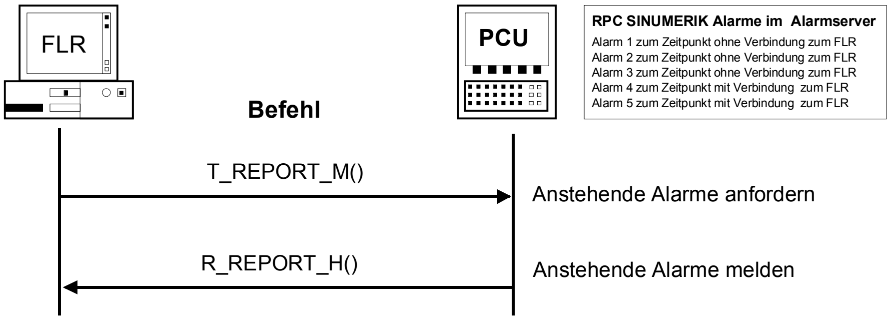


Bild 5-3 Anstehende Alarme anfordern

T_REPORT_M (Host,
Machine,
OrderNum,
Übertragungsrichtung **FLR** → **SINUMERIK**

Bedeutung

Mit T_REPORT_M können im Alarmserver anstehende Alarme und Meldungen unter Berücksichtigung des Meldungsfilters angefordert werden. Es werden **alle** anstehenden Alarme und Meldungen des Alarmservers angefordert, unabhängig davon, ob sie zu einem Zeitpunkt mit oder ohne Verbindung zum FLR entstanden. Die Funktion ermöglicht somit eine Synchronisation zwischen FLR und SINUMERIK bezüglich der Alarme und Meldungen.
Andernfalls werden nur solche Alarme an den FLR weitergegeben, die zu einem Zeitpunkt **mit** Verbindung zum FLR entstanden.
Die mit **T_REPORT_M** angeforderten Alarme und Meldungen werden mit **R_REPORT_H** gemeldet.

Daten

Tabelle 5-6 Parameter der Alarmanforderung

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)

5.6 Austausch von Bedienernachrichten

Die beiden RPC-Funktionen **R_MESSAGE_M** und **R_MESSAGE_H** dienen zum Austausch von Klartext-Meldungen zwischen FLR und SINUMERIK.

Die Anzeige der an der SINUMERIK ankommenden Meldungen vom FLR (**R_MESSAGE_M**) erfolgt über den Alarmserver (im Header). Die Priorität der Meldungen kann im FLR mit Hilfe des Parameters **ResInt1** von R_MESSAGE_M gesetzt werden. Der Parameter **ResInt1** kann folgende Bedeutung im Zusammenhang mit R_MESSAGE_M haben:

ResInt1 = 0 : Alarm-Priorität mit dem Standardwert 100
<> 0 : Alarm-Priorität

Hinweis

Die Alarmpriorität beeinflusst die Position in der Alarmliste (siehe Alarmserver Beschreibung).

5.6.1 Nachricht an die SINUMERIK

Funktionsaufruf

R_MESSAGE_M(Host,
Machine,
OrderNum,
Message,
Resint1,
Resint2,
Resbyte)

Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Nachricht an der Bedientafel der SINUMERIK anzeigen

Daten

Tabelle 5-7 Parameter der Nachricht an SINUMERIK

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
Message	Nachrichtentext	128 Byte (string)
Resint1	Reserve 1	4 Byte (long int.)
Resint2	Reserve 2	4 Byte (long int.)
Resbyte	Reserve 3	8 Byte (string)

Anwendungshinweise

Vom FLR werden z.B. Nachrichten über bereitzustellende Werkzeuge oder Materialien an den Maschinenbediener übertragen, um die Arbeitsvoraussetzungen für die nächste Teileprogrammbearbeitung zu schaffen.

Hinweis

Die Anzeige erfolgt über die Alarmzeile im Header

Beispiel

```
R_MESSAGE_M("FLR1", "BAZ3", 0, "Hallo Maschine", 0, 0, "\0");
```

5.6.2 Nachricht an den FLR

Aufgerufene Funktion

R_MESSAGE_H(Host,
Machine,
OrderNum,
Message,
Resint1,
Resint2,
Resbyte)

Übertragungsrichtung **SINUMERIK → FLR**

Bedeutung

Nachricht am Fertigungsleitrechner anzeigen.

Daten

Tabelle 5-8 Parameter der Nachricht an FLR

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
Message	Nachrichtentext	128 Byte (string)
Resint1	Reserve 1	4 Byte (long int.)
Resint2	Reserve 2	4 Byte (long int.)
Resbyte	Reserve 3	8 Byte (string)

Anwendungshinweise

Der Maschinenbediener teilt dem FLR z.B. die abgeschlossenen Vorarbeiten für eine Programmbearbeitung mit.

Hinweis

Die Texteingabe erfolgt über einen Dialog im Dialogprogramm von RPC SINUMERIK.

5.7 Datendialoge

Beschreibung

Größere Datenmengen, wie NC-Programme oder Werkzeugdaten, werden als Datei übertragen. Da bei Einsatz von FTP (File Transfer Protokoll) die SINUMERIK nur FTP-Client sein kann, muss die Dateiübertragung immer von RPC SINUMERIK in der SINUMERIK ausgeführt werden. Der FLR teilt mit dem RPC-Call **R_DATA_M** der RPC SINUMERIK Rechnerkopplungssoftware in der SINUMERIK mit, dass eine Datei zur Verarbeitung bereitliegt. Die Rechnerkopplungssoftware holt sich daraufhin die Datei und verarbeitet sie. In der anderen Richtung überträgt RPC SINUMERIK in der SINUMERIK die Datei und informiert mit dem RPC-Call **R_DATA_H** den FLR, dass eine Datei zur Verarbeitung auf dem FLR bereit liegt. Datendialoge beziehen sich auf die mit SCCONFIG projektierten PUT_/GET-Verzeichnisse. Pfadnamen müssen immer aus "Sicht" von RPC SINUMERIK angegeben werden. Es können sowohl freigegebene Laufwerksbezeichnungen (Shares) als auch die UNC-Notation verwendet werden. Wenn die Dateien in den projektierten Verzeichnissen liegen, kann beim Parameter Name2 der Dateiname ohne Pfad angegeben werden. Bei FTP hängen die Pfadangaben von den Möglichkeiten des FTP-Servers ab.

Programmübertragung

NC-Programme werden als Dateien übertragen. NC-Programme müssen nach ihrer Übertragung in die NC-Datenhaltung übernommen werden.

Kennzeichnung der Dateiinhalte

Bestandteil der Daten in den Aufrufen R_DATA_M() und R_DATA_H() ist eine Unterfunktionsnummer SFkt, die die Art der angebotenen Daten und damit die Angaben zur ordnungsgemäßen Übernahme enthält.
Eine Datei mit Werkzeugdaten muss werkzeugweise gelesen und entsprechend der angegebenen Unterfunktionsnummer verarbeitet werden. Die Dateien sind nach ihrer Abarbeitung von der Verarbeitungsfunktion zu löschen.
Beim Start der Rechnerkopplung sind alle alten empfangenen Dateien zu löschen.
Die Unterfunktionsnummern sind gleich bei:

- Anforderung,
- Übernahme und
- Löschen.

Tabelle 5-9 Unterfunktionsnummern: SFkt Datendialoge

Unter-fkt.-Nr.	Funktion	Bemerkung
1	NC-Programm	Name1 = NC-Programm Name2 = Dateiname mit Pfad auf dem FLR
10	Liste der vorhandenen NC-Programme	Name1 = Datenhaltungspfad Name2 = Name der Listendatei
20	Werkzeugzustandsdaten **aller** WZ **Vollständiger** WZ-Datensatz	Name1 = leer Name2 = Dateiname mit Pfad auf dem FLR
21	Werkzeugzustandsdaten eines WZ Variante 1: **Vollständiger** WZ-Datensatz	Name1 = Identnummer, Duplonummer Name2 = Dateiname mit Pfad auf dem FLR
22	Werkzeugzustandsdaten eines WZ Variante 2: **reduzierter** WZ-Datensatz	Name1 = Identnummer, Duplonummer Name2 = Dateiname mit Pfad auf dem FLR
23	Werkzeugzustandsdaten eines WZ Variante 3: **reduzierter** WZ-Datensatz	Name1 = Identnummer, Duplonummer Name2 = Dateiname mit Pfad auf dem FLR
24	WZ-Daten eines WZ mit Adapternummer **Vollständiger** WZ-Datensatz	Name1 = Adapternummer Name2 = Dateiname mit Pfad auf dem FLR
26	Wahlfreies- /manuelles Beladen eines WZ **Vollständiger** WZ-Datensatz	Name1 = Identnummer, Duplonummer Name2 = Dateiname mit Pfad auf dem FLR
27	Wahlfreies-/manuelles Entladen eines WZ	Name1 = Identnummer, Duplonummer Name2 = Dateiname mit Pfad auf dem FLR
28	WZ Beladen von WZ-Palette	Name1 = Werkzeugpalettennummer Name2 = Name d. Datei mit WZ-Zust. Daten
29	WZ Entladen auf WZ Palette	Name1 = Werkzeugpalettennummer Name2 = Name d. Datei mit WZ-Zust. Daten
50	Maschinenbelegungsdaten	Name1 = leer, Name2 = Dateiname mit Pfad auf dem FLR
90	Beliebige Datei übertragen Keine weitere Bearbeitung - optional	Name1 = Quelldateiname mit Pfad Name2 = Zieldateiname mit Pfad

5.7.1 Daten als Datei von der SINUMERIK anfordern

Funktionsaufruf

T_DATA_M (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Daten als Datei von der SINUMERIK anfordern.

Daten

Tabelle 5-10

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
SFkt	Unterfunktionsnummer	4 Byte (long int.)
Name1	Name siehe Tabelle 5-9 Unterfunktionsnummern:SFkt	128 Byte (string)
Name2	zusätzlicher Name siehe Tabelle 5-9: Unterfunktionsnummern:SFkt	128 Byte (string)

Beispiel

```
T_DATA_M ("FLR1", "BAZ3", 0, 1, "\mpf.dir\ Kw15.mpf", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 10, "\mpf.dir", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 20, "\0", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 21, "Bohrer10mm,0002", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 22, "Bohrer10mm,0002", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 23, "Bohrer10mm,0002", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 50, "\0", "\0");
T_DATA_M ("FLR1", "BAZ3", 0, 90, "f:\add_on\mcis_rpc.log", "\0");
```

5.7.2 Daten als Datei vom Fertigungsleitrechner anfordern

Aufgerufene Funktion

T_DATA_H (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2)

Übertragungsrichtung: **SINUMERIK → FLR**

Bedeutung

Daten als Datei vom Fertigungsleitrechner anfordern.

Daten

Tabelle 5-11 Parameter zur Datenanforderung

Parameter	**Beschreibung**	**Format**
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
SFkt	Unterfunktionsnummer	4 Byte (long int.)
Name1	Name siehe Tabelle 5-9 Unterfunktionsnummern:SFkt	128 Byte (string)
Name2	zusätzlicher Name siehe Tabelle 5-9 Unterfunktionsnummern:SFkt	128 Byte (string)

5.8 Übertragene Dateien übernehmen

5.8.1 Auftrag an Maschine: Daten übernehmen

Funktionsaufruf

R_DATA_M (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2,
Date,
LastFile)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Der FLR fordert RPC SINUMERIK auf, die angegebene Datei vom FLR zu holen und zu verarbeiten (z.B. in die Datenhaltung zu übernehmen).

Daten

Tabelle 5-12 Parameter zur Übernahmeaufforderung

<table>
<tr><th>Parameter</th><th colspan="3">Beschreibung</th><th>Format</th></tr>
<tr><td>Host</td><td colspan="3">Fertigungsleitrechnername (Host)</td><td>16 Byte (string)</td></tr>
<tr><td>Machine</td><td colspan="3">Maschinenname</td><td>16 Byte (string)</td></tr>
<tr><td>OrderNum</td><td colspan="3">Auftragsnummer</td><td>4 Byte (long int.)</td></tr>
<tr><td>SFkt</td><td colspan="3">Unterfunktionsnummer</td><td>4 Byte (long int.)</td></tr>
<tr><td>Name1</td><td colspan="3">Name , siehe Tabelle 5-9 Unterfunktionsnummern:SFkt</td><td>128 Byte (string)</td></tr>
<tr><td>Name2</td><td colspan="3">zusätzlicher Name, siehe Tabelle 5-9 Unterfunktionsnummern:SFkt</td><td>128 Byte (string)</td></tr>
<tr><td>Date</td><td colspan="3">Datum der letzten Änderung
(Unix-Uhrzeit, nur für NC-Programm-Datei)</td><td>4 Byte (long int.)</td></tr>
<tr><td rowspan="5">LastFile</td><td colspan="3">Letzte Datei eines Werkstückes</td><td rowspan="5">4 Byte (long int.)</td></tr>
<tr><td>Bit 7</td><td>Bit 0</td><td>Funktion</td></tr>
<tr><td>0</td><td>0</td><td>es folgen weitere Dateien</td></tr>
<tr><td>0</td><td>1</td><td>letzte Datei eines Werkstückes oder einzelne Datei</td></tr>
<tr><td>1</td><td>0</td><td>Anforderungskennung zurücksetzen</td></tr>
</table>

Anwendungshinweis

Bit 7 ist nur von Bedeutung wenn über die RK840D Bedienoberfläche ein NC-Programm beim FLR angefordert wird, vom FLR jedoch bewusst ein NC-Programm mit anderem Namen gesendet wird. Standardmäßig wird die RPC SINUMERIK-interne Anforderungskennung nur bei gleicher SFkt und Name1 zurückgesetzt, über Bit 7 kann der FLR die Rücksetzung der Anforderungskennung auslösen.

Beispiel

R_DATA_M ("FLR1", "BAZ3", 0, 1, "\mpf.dir\ Kw15.mpf", "f:\ncpro\NCKW0815.txt", 862826400, 1);

R_DATA_M ("FLR1", "BAZ3", 0, 1, "WKS.DIR\Zylinderkopf.wpd\ Kw15.mpf", "f:\ncpro\NCKW0815.txt", 862826400, 1);
R_DATA_M ("FLR1", "BAZ3", 0, 10, "Hauptprogramme", "f:\tmp\NCListe.txt", 0, 1);
R_DATA_M ("FLR1", "BAZ3", 0, 26, "Bohrer10mm,0002", "f:\tmp\wzfile.txt");
R_DATA_M ("FLR1", "BAZ3", 0, 27, "TP003", " f:\tmp\tp003.txt ");
R_DATA_M ("FLR1", "BAZ3", 0, 28, "TP003", " f:\tmp\tp003.txt ");
R_DATA_M ("FLR1", "BAZ3", 0, 1001, "c:\mmc2\oemdata.txt", "c:\tmp\oemdata.txt");

R_DATA_M ("FLR1", "BAZ3", 0, 1, "WKS.DIR\Zylinderkopf.wpd\ Kw15.mpf", "NCKW0815.txt", 862826400, 1);

Synchron während R_DATA_M wird die Datei vom FLR geholt, die Verarbeitung kann jedoch nicht synchron im RPC erfolgen. Deshalb kann der Returnwert des RPC nur quittieren, ob die Übernahme der Datei erfolgreich war oder nicht. RPC SINUMERIK sendet nach der Verarbeitung der Datei R_REPORT_H mit Typ=5 und Fehlernummer = SFkt von R_DATA_M als positive Quittung.

5.8.2 Auftrag an FLR: Daten übernehmen

Aufgerufene Funktion

R_DATA_H (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2,
Date,
LastFile)

Übertragungsrichtung: **SINUMERIK → FLR**

Bedeutung

Die mit Filetransfer bereits überspielte Datei soll in die Datenhaltung des Fertigungsleitrechners übernommen werden.

Daten

Tabelle 5-13 Parameter zur Übernahmeaufforderung

Parameter	**Beschreibung**	**Format**
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
SFkt	Unterfunktionsnummer	4 Byte (long int.)
Name1	Name siehe: Tabelle 5-9 Unterfunktionsnummern:SFkt	128 Byte (string)
Name2	zusätzlicher Name, siehe: Tabelle 5-9 Unterfunktionsnummern:SFkt	128 Byte (string)
Date	Datum der letzten Änderung (Unix-Uhrzeit, nur für NC-Programm-Datei)	4 Byte (long int.)
LastFile	Letzte Datei eines Werkstückes 0: es folgen weitere Dateien 1: letzte Datei eines Werkstückes oder einzelne Datei	4 Byte (long int.)

5.9 Daten löschen

5.9.1 Daten auf der SINUMERIK löschen

Funktionsaufruf

C_DELETE_M (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Daten der vorangegangen Filetransferübertragung löschen.

Daten

Tabelle 5-14 Parameter zur Löschaufforderung

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
SFkt	Unterfunktionsnummer	4 Byte (long int.)
Name1	Name der zu löschenden Datei	128 Byte (string)
Name2	zusätzlicher Name	128 Byte (string)

Hinweis

Zur Zeit können nur Dateien in der Datenhaltung angesprochen werden,
SFkt = 1
z.B.:Name1 =“\mpf.dir\zylinderkopf.mpf“

Beispiel

C_DELETE_M ("FLR1", "BAZ3", 0, 1, "\mpf.dir\ Kw15.mpf", "\0");

5.10 NC-Programme

Die folgende Darstellung zur Behandlung der Programme ist eine spezielle Anwendung der in **Kapitel 5.7** beschriebenen Funktionen. Deren Kenntnis wird hier vorausgesetzt.

5.10.1 NC-Programm anfordern, Initiative durch FLR

1. Funktionsaufruf

T_DATA_M ()
SFkt = 1
Name1 = Programmname in der Datenhaltung, z.B.: \mpf.dir\Halter4711.mpf
Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Der Fertigungsleitrechner fordert von der SINUMERIK ein bestimmtes NC-Programm an.

2. Dateitransfer

Die Datei mit angefordertem NC-Programm wird übertragen.

3. Aufgerufene Funktion

R_DATA_H()
SFkt = 1
Name1 = Programmname in der Datenhaltung
Name2 = Name incl. Pfad der Datei auf dem FLR
Date = Datum der letzten Änderung
Übertragungsrichtung **SINUMERIK → FLR**

Bedeutung

Die SINUMERIK fordert vom Fertigungsleitrechner ein bestimmtes NC-Programm an.

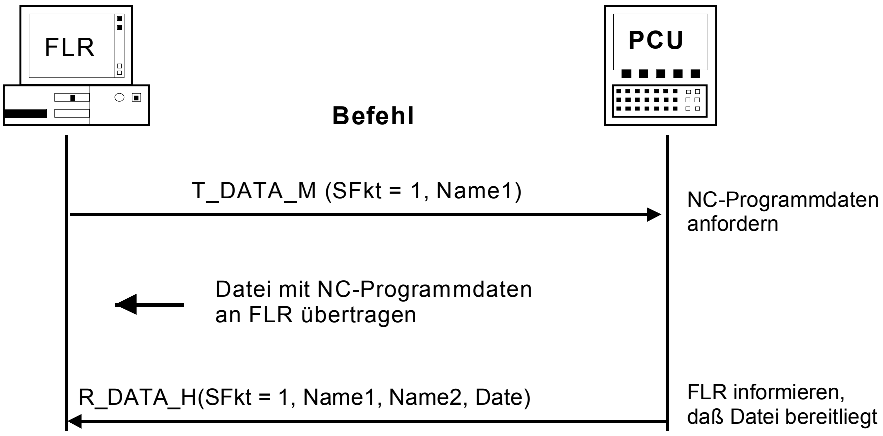


Bild 5-4 NC-Programm anfordern, Initiative durch FLR

5.10.2 NC-Programm anfordern, Initiative durch die SINUMERIK

1. Aufgerufene Funktion

T_DATA_H ()
SFkt = 1
Name 1 = Programmname in der Datenhaltung
Übertragungsrichtung **FLR -> SINUMERIK**

Bedeutung

Die SINUMERIK fordert vom Fertigungsleitrechner ein bestimmtes NC- Programm an.

2. Funktionsaufruf

R_DATA_M()
SFkt = 1
Name1 = Programmname in der Datenhaltung
Name2 = Name incl. Pfad der Datei auf dem FLR
Date = Datum der letzten Änderung
Übertragungsrichtung **FLR -> SINUMERIK**

Bedeutung

Der Fertigungsleitrechner stellt für die SINUMERIK das angeforderte NC-Programm bereit.

3. Dateitransfer

RPC SINUMERIK übernimmt die Datei des angeforderten NC-Programms in die SINUMERIK und in die Datenhaltung

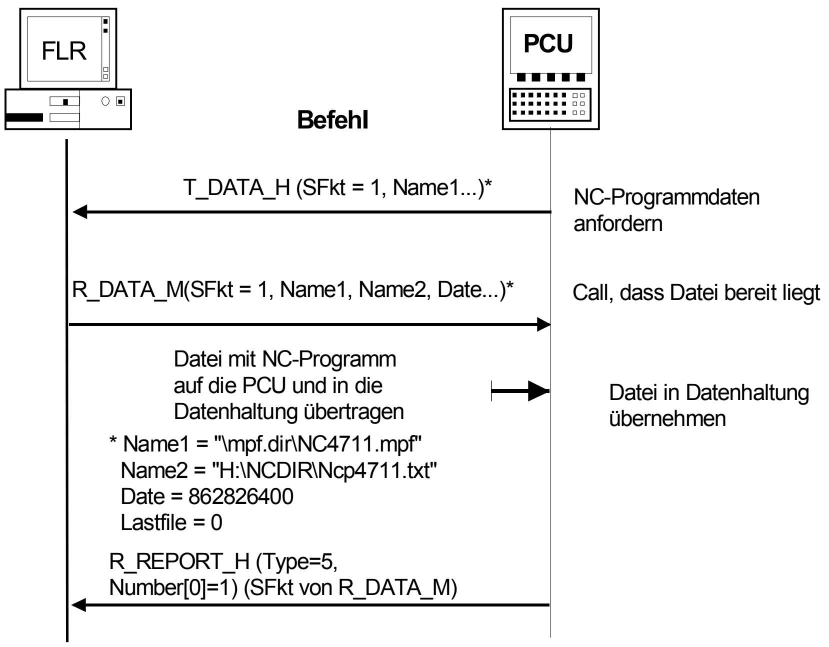


Bild 5-5 NC-Programm anfordern, Initiative durch die SINUMERIK

Hinweis

Es können nur einzelne NC-Programme angefordert und übertragen werden.

Die Werkstückträgerbezeichnung muss mit '\0' abgeschlossen werden und darf inclusive '\0' maximal 6 Byte lang sein.
Beim NC-Programm muss immer der Datenhaltungspfad enthalten sein.
z.B: NCProg = "\mpf.dir\Zylinderkopf.mpf"

Da die Datenhaltung in der SINUMERIK keine Versionen kennt, kann das letzte Änderungsdatum und die Dateigröße angegeben werden. Ist in der Datenhaltung der SINUMERIK das NC-Programm vorhanden, aber mit einer anderen Größe bzw. einem anderen Änderungsdatum, so muss RPC SINUMERIK die aktuelle Datei vom FLR anfordern, bevor das Programm für die Bearbeitung aktiviert werden darf.

5.10.3 NC-Programm übertragen

Funktionsaufruf

R_DATA_M ()
SFkt = 1
Name1 = Programmname in der Datenhaltung
Name2 = Name incl. Pfad der Datei auf dem FLR
Date = Datum der letzten Änderung
Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Der Fertigungsleitrechner stellt für die SINUMERIK ein bestimmtes NC-Programm bereit.

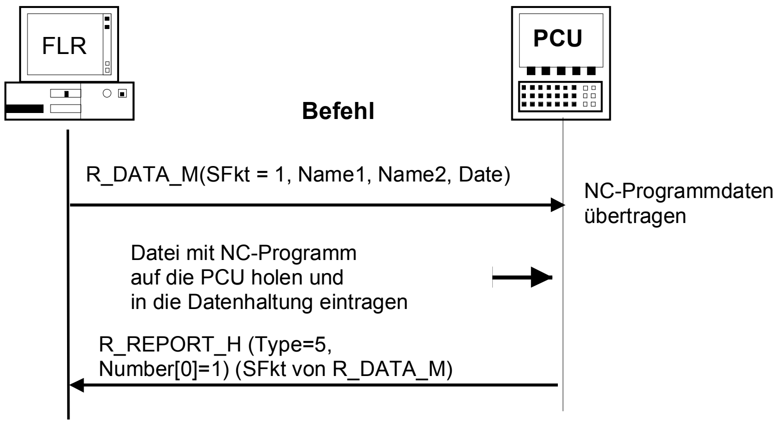


Bild 5-6 NC-Programm übertragen, Initiative durch den FLR

Aufgerufene Funktion

R_DATA_H()
SFkt = 1
Name1 = Programmname in der Datenhaltung
Name2 = Name incl. Pfad der Datei auf dem FLR
Date = Datum der letzten Änderung

Übertragungsrichtung **SINUMERIK → FLR**

Bedeutung

Die SINUMERIK stellt dem Fertigungsleitrechner ein bestimmtes NC-Programm bereit.

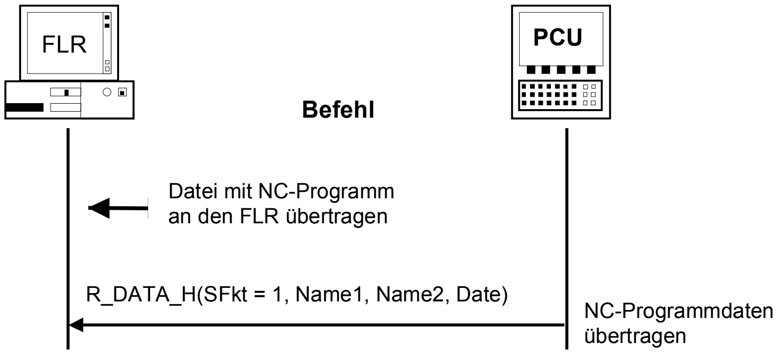


Bild 5-7 NC-Programm übertragen, Initiative durch die SINUMERIK

5.10.4 Programme auf Maschine löschen

Funktionsaufruf

C_DELETE_M (SFkt = 1, Name1)
Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Der Fertigungsleitrechner weist die SINUMERIK an, das durch Name1 spezifizierte Programm zu löschen.

Daten

Die zu übergebenden Daten sind oben unter C_DELETE_M() beschrieben. Zu besetzen sind dabei:
SFkt = 1, Name1 = "\mpf.dir\zylinderkopf.mpf" oder
"\spf.dir\4711.spf"

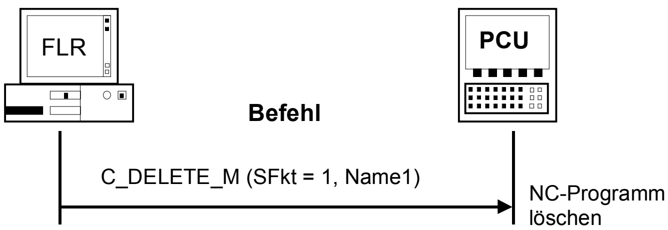


Bild 5-8 NC-Programm löschen, Initiative durch FLR

5.10.5 Liste der vorhandenen NC-Programme anfordern, Initiative durch FLR

1. Funktionsaufruf

T_DATA_M() mit
SFkt=10
Name1 = Pfad in der Datenhaltung (z.B. "\mpf.dir")
Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Anforderung des Fertigungsleitrechners an die Maschine, die Liste der vorhandenen Programme zu übertragen.

2. Dateitransfer

Datei mit NC-Programmliste wird an den FLR übertragen

3. Aufgerufene Funktion

R_DATA_H() mit
SFkt=10
Name1 = Pfad der Datenhaltung
Name2 = Dateiname mit NC-Programmliste
Übertragungsrichtung **SINUMERIK → FLR**

Bedeutung

Übertragung der Maschine an den Fertigungsleitrechner: Liste der vorhandenen NC-Programme.

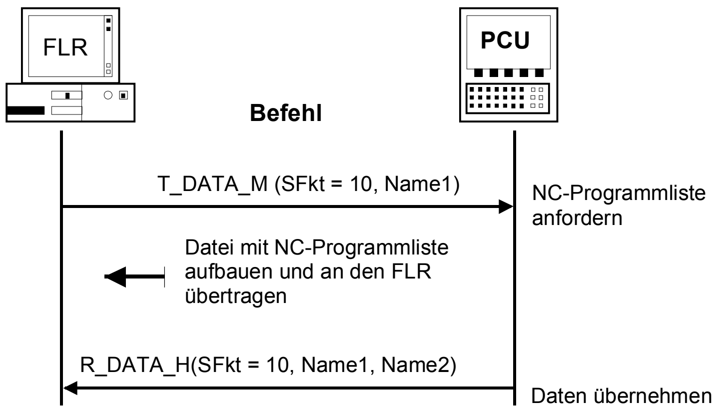


Bild 5-9 Liste der vorhandenen NC-Programme anfordern, Initiative durch FLR

5.10.6 Liste der vorhandenen NC-Programme anfordern, Initiative durch die SINUMERIK

1. Aufgerufene Funktion

T_DATA_H ()
SFkt = 10
Name 1 = Pfad in der Datenhaltung (z.B. “\mpf.dir“)
Übertragungsrichtung **FLR** →**SINUMERIK**

Bedeutung

Anforderung der Maschine an den Fertigungsleitrechner, die Liste der vorhandenen NC-Programme zu übertragen.

2. Funktionsaufruf

R_DATA_M ()
SFkt = 10
Name1 = Pfad in der Datenhaltung
Name2 = Dateiname mit NC-Programmliste

Übertragungsrichtung **FLR** → **SINUMERIK**

Bedeutung

Aufforderung an die Maschine, die bereitgestellte NC- Programmliste zu übernehmen.

3. Dateitransfer

RPC SINUMERIK übernimmt die Datei mit der NC-Programmliste auf die SINUMERIK.

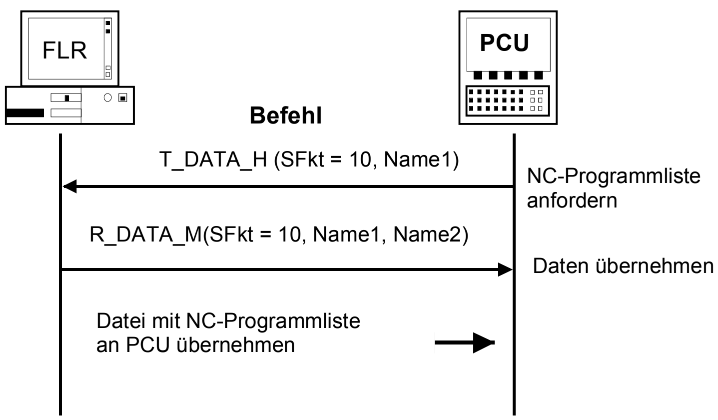


Bild 5-10 Liste der vorhandenen NC-Programme anfordern, Initiative durch die SINUMERIK

5.10.7 NC-Programmliste übertragen

Funktionsaufruf

R_DATA_M() mit
SFkt=10
Name1 = Pfad der Datenhaltung für
Name2 = Dateiname mit NC-Programmliste
Übertragungsrichtung **FLR → SINUMERIK**

Bedeutung

Übertragung des Fertigungsleitrechners an die Maschine: Liste der vorhandenen Programme

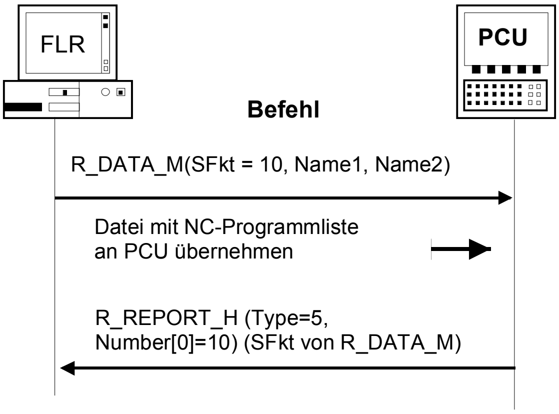


Bild 5-11 NC-Programmliste übertragen

Aufgerufene Funktion

R_DATA_H() mit
SFkt=10
Name1 = Pfad der Datenhaltung
Name2 = Dateiname mit NC-Programmliste
Übertragungsrichtung **SINUMERIK → FLR**

Bedeutung

Übertragung von der Maschine an den Fertigungsleitrechner: Liste der vorhandenen Programme

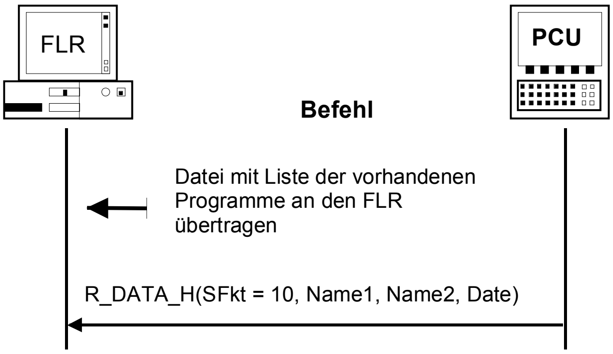


Bild 5-12 NC-Programmliste zum FLR übertragen

Die Datei mit der NC-Programmliste enthält eine Auflistung der Dateien sowie in diesem Datenhaltungspfad enthaltener Unterverzeichnisse. Der Inhalt dieser Unterverzeichnisse wird nicht aufgelistet, er muss bei Bedarf mit einer eigenen Anforderung ermittelt werden.

Aufbau der mit R_DATA_H zurückgelieferten Datei

\mpf.dir
Zylinderkopf.MPF,FM,5320,876403708
Kurbelwelle.MPF,FN,8300,862826400

Zeilen

In der ersten Zeile der Datei steht das Verzeichnis, das bei Name1 angegeben wurde und dessen Inhalt aufgelistet wird. In jeder weiteren Zeile steht ein Datei- oder Subdirectoryname mit den Zusatzinformationen, die durch Komma getrennt sind.

Spalten

In der ersten Spalte steht der NC-Programmname bzw. der Name des Subdirectories.
In der zweiten Spalten stehen zwei Zeichen, das

- erste kennzeichnet ob sich um eine Datei (F) oder um ein Subdirectory (D) handelt
- das zweite Zeichen zeigt ob die Datei in der PCU oder im NCK liegt

Beispiele:
"Fx" - File
"Dx" - Verzeichnis
"xM" - liegt auf PCU
"xN" - liegt im NCK o. in NCK und PCU

Die dritte Spalte enthält die Dateigröße in Byte.

Die vierte Spalte enthält als Dezimalzahl das **Datum** der Datei als UNIX-Uhrzeit in Sekunden seit dem 01.01.1970.

Die Zeit bei Zylinderkopf 876403715 steht für 09.10.1997 15:28:35, bei Kurbelwelle 862826400 für 05.05.1997 12:00:00.

5.11 Werkzeugdialoge

Werkzeugdaten werden immer als Datei übertragen.
Der Dateiaufbau ist unter **Kapitel 4.1** beschrieben.

5.11.1 Gesamte Werkzeugmagazinbelegung abfragen

Übertragungs-richtung	**Befehl**		**Bedeutung**
FLR → SINUMERIK	T_DATA_M	(SFkt = 20)	Alle Werkzeugdaten anfordern
FLR ← SINUMERIK			Datei mit allen WZ-Zustands-daten an den FLR übertragen
FLR ← SINUMERIK	R_DATA_H	(SFkt = 20, Name1 = leer, Name2= Datei mit allen WZ-Zustandsdaten)	Ankunft der Datei dem FLR mitteilen

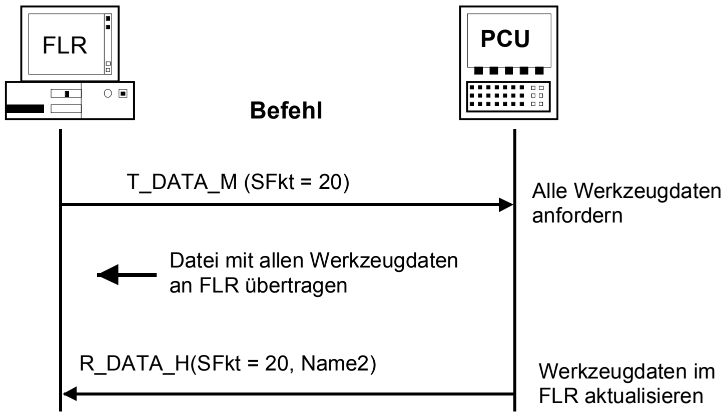


Bild 5-13 Abfrage aller Werkzeuge, Initiative durch den FLR

5.11.2 WZ-Daten mit WZ-Adapternummer (Option)

Übertragungs-richtung	Befehl		Bedeutung
FLR ← SINUMERIK	T_DATA_H	(SFkt = 24, Name1=Adapter-nummer)	Werkzeugdaten mit Adapternummer anfordern
FLR → SINUMERIK	R_DATA_M	(SFkt = 24, Name1=Adapternummer, Name2=Dateiname mit WZ-Zustandsdaten)	WZ-Daten mit Adapternummer übertragen

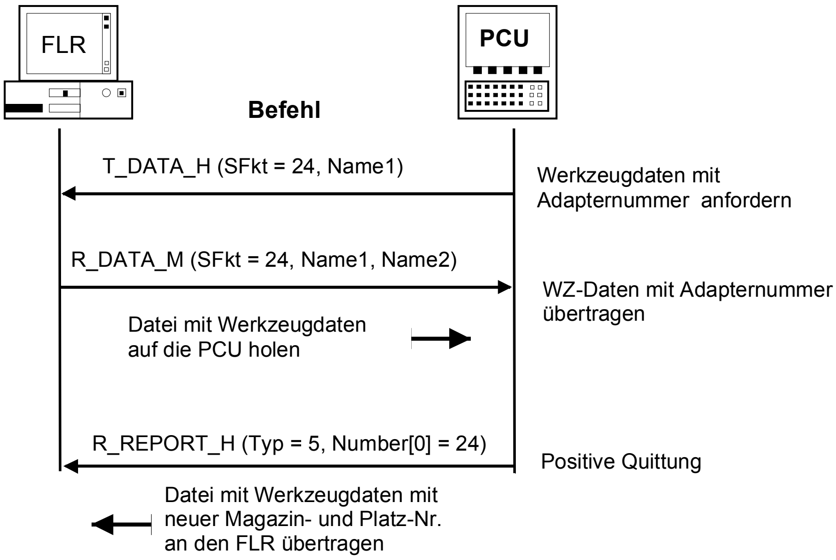


Bild 5-14 WZ-Daten mit WZ-Adapternummer (Option)

5.11.3 Wahlfreies-/ manuelles Beladen

Übertragungs-richtung	**Befehl**		**Bedeutung**
FLR ← SINUMERIK	T_DATA_H	(SFkt = 26, Name1= Identnr, Duplonr)	Werkzeugdaten anfordern
FLR → SINUMERIK	R_DATA_M	(SFkt = 26, Name1=Identnr, Duplonr, Name2=Dateiname mit WZ-Zustandsdaten)	Aufforderung an die SINUMERIK, Datei zu holen. Datei mit WZ-Zustandsdaten auf die SINUMERIK holen. Werkzeug beladen
FLR ← SINUMERIK	R_DATA_H	(SFkt = 21, Name1= Identnr, Duplonr Name2=Dateiname mit WZ-Zustandsdaten)	Datei mit WZ-Zustandsdaten mit neuer Magazin- und Platz-Nr. an den FLR (das ist die Beladequittung)

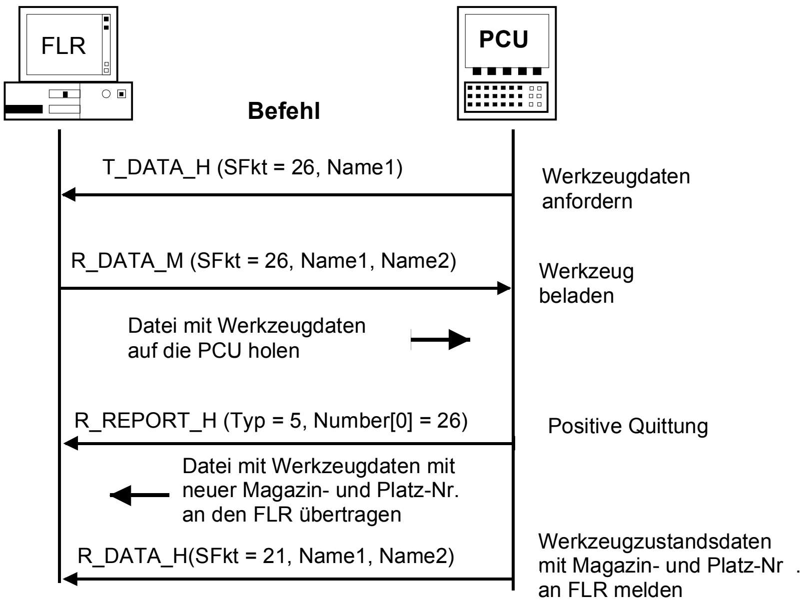


Bild 5-15 Werkzeug beladen, Initiative durch die SINUMERIK

5.11.4 Wahlfreies-/ manuelles Entladen

Übertragungs-richtung	**Befehl**		**Bedeutung**
FLR ← SINUMERIK	R_DATA_H	(SFkt = 27, Name1=Identnr, Duplonr, Name2=Dateiname mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten mit neuer Magazin- und Platznummer an FLR (Entladequittung)

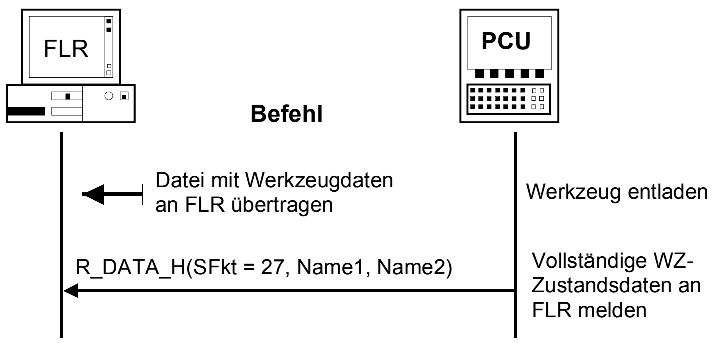


Bild 5-16 Werkzeug entladen, Initiative durch die SINUMERIK

5.11.5 Werkzeug melden

Übertragungs-richtung	Befehl		Bedeutung
FLR ← SINUMERIK	R_DATA_H	(SFkt = 21, Name1=Identnr, Duplonr, Name2=Dateiname mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten an FLR (Werkzeug melden)

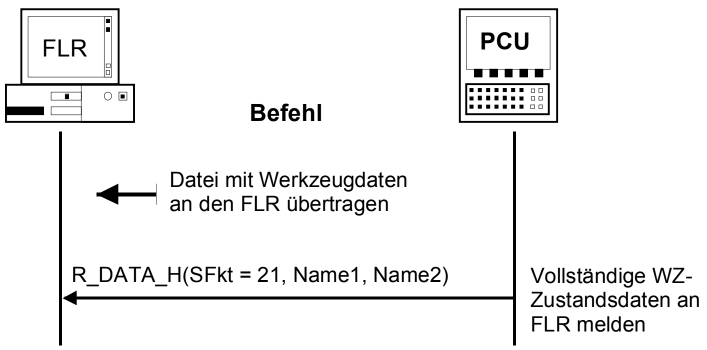


Bild 5-17 Werkzeug melden, Initiative durch die SINUMERIK

5.11.6 Beladen von Werkzeugpalette / -kassette (Option)

Übertragungs-richtung	Befehl		Bedeutung
FLR → SINUMERIK	R_DATA_M	(SFkt = 27, Name1=Werkzeug-palettennummer, Name2=Datei mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten auf die SINUMERIK holen
FLR ← SINUMERIK	R_DATA_H	(SFkt = 20, Name1= Identnr, Duplonr, Name2= Dateiname mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten mit neuer Magazin- und Platz-Nr.an den FLR

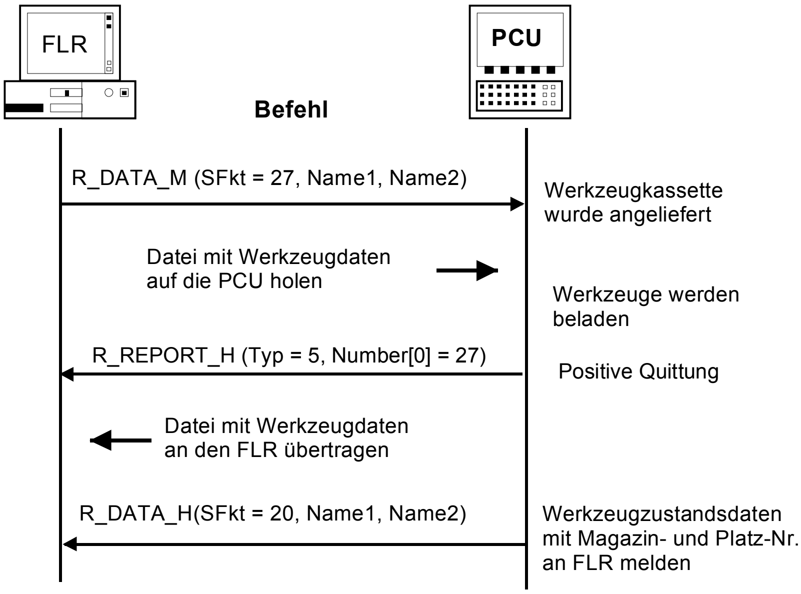


Bild 5-18 Beladen von Werkzeugpalette/-kasette

5.11.7 Entladen von Werkzeugpalette / -kassette (Option)

Übertragungs-richtung	Befehl		Bedeutung
FLR → SINUMERIK	R_DATA_M	(SFkt = 28, Name1=Werkzeug-paletten-nummer, Name2=Datei mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten auf die SINUMERIK holen Wrkzeuge werden entladen
FLR ← SINUMERIK	R_DATA_H	(SFkt = 20, Name1= Identnr, Duplonr, Name2=Dateiname mit WZ-Zustandsdaten)	Datei mit Werkzeug-Zustandsdaten mit neuer Magazin- und Platz-Nr.an den FLR

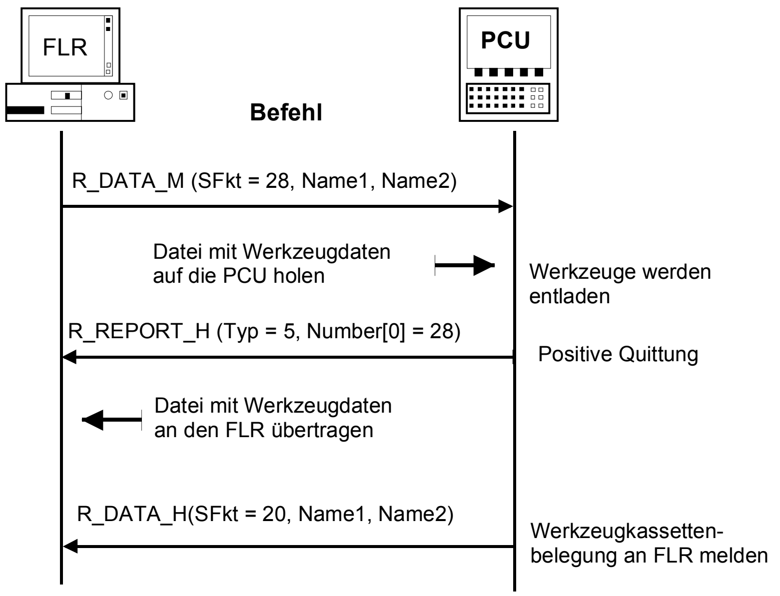


Bild 5-19 Entladen von Werkzeugpalette/-kasette

5.12 Maschinenbelegungsdaten

Bei Maschinen mit mehr als 3 Haltestellen/Liegeplätzen, wie z.B. Transportsystem mit Ablageplätzen, werden die Daten aller Haltestellen/Liegeplätze in einer ASCII-Datei übertragen. Nach Übertragung der Datei an den FLR folgt der RPC-**R_DATA_H (SFkt = 50)**
(siehe **Kapitel 5.8.2**).
Pro Haltestelle/Liegeplatz wird,

- die Haltestellennummer
- die Werkstückträgernummer und
- der Werkstückträgerzustand

übertragen.
Die Datenfelder stehen in der Datei mit **Komma** getrennt in folgender Weise:

Aufbau der ASCII-Datei

DockPos 1, DockPosStatus, WPC, WPCStatus **<Zeilenvorschub>**
DockPos 2, DockPosStatus, WPC, WPCStatus **<Zeilenvorschub>**
...
DockPos n, DockPosStatus, WPC, WPCStatus **<EOF>**

Tabelle 5-15 Beschreibung der Datei-Parameter

Parameter	**Beschreibung**	**Format**
Dockpos	Haltestellennummer	ASCII
DockPosStatus	Haltestellenzustand 0: freigegeben 1: für TPS gesperrt 2: gestört	ASCII
WPC	Werkstückträgerbezeichnung	ASCII
WPCStatus	Werkstückträgerzustand 1: unbearbeitet, ohne Programmzuordnung 2: unbearbeitet, Programmzuordnung erfolgt 16: in Bearbeitung 32: fertig bearbeitet 64: bearbeitet mit Fehlern 128: nur zur Pufferung	ASCII

Der FLR kann diese Daten mit T_DATA_M(SFkt = 50) anfordern.

5.13 Allgemeine Auftragsfunktion

C_ORDER_M (Host,
Machine,
OrderNum,
SFkt,
Name1,
Name2,
Name3,
Name4,
Parameter1,
Parameter2,
Parameter3,
Parameter4)

Übertragungsrichtung: **FLR → SINUMERIK**

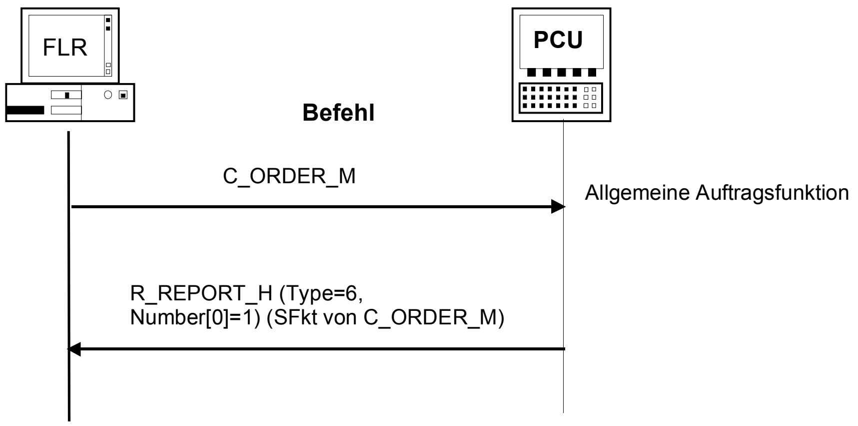


Bild 5-20 Allgemeine Auftragsfunktion

Bedeutung

Nach der Abarbeitung sendet RPC SINUMERIK entweder als positive Quittung R_REPORT_H mit Typ= 6 und Fehlernummer=SFkt von C_ORDER_M oder als negative Quittung R_REPORT_H mit Typ= 4 und mit einer entsprechenden Fehlernummer.

Tabelle 5-16 Unterfunktionsnummer SFkt für die Allgemeine Auftragsfunktion

Unterfkt.-Nr. (SFkt)	**Funktion**	**Bemerkung**	
2	NC-Programm laden	Name1 = Name2 = Parameter1=	NC-Programmname NCK-Name (optional) Funktion (0,2,3)
3	Laden und Anwählen von NC-Programmen	Name1 = Name2 = Parameter1= Parameter2=	NC-Programmname NCK-Name (optional) Funktion (0-3) Kanalnummer
4	Entladen von NC-Programmen	Name1 = Name2 =	NC-Programm NCK-Name
5	Anwählen von NC-Programmen	Name1 = Name2 = Parameter1= Parameter2=	NC-Programmname NCK-Name (optional) Funktion (0,1) Kanalnummer
6	Abwählen von NC-Programmen	Name2 = Parameter2=	NCK-Name (optional) Kanalnummer
100	Datum/Uhrzeit auf SINUMERIK/PLC setzen	Name2 = Parameter1 Parameter2=	NCK-Name (optional) UNIX-Zeit Datum / Uhrzeit SINUMERIK / PLC

Hinweis

Der NCK-Name kann erst nach dem Eintrag der notwendigen Erweiterungen in die Dateien NETNAMES.INI und MMC.INI verwendet werden.

Daten

Tabelle 5-17 Parameter Mode-Umschaltung

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
SFkt	Unterfunktionsnummer	4 Byte (long int.)
Name1	*	128 Byte (string)
Name2	*	128 Byte (string)
Name3	*	128 Byte (string)
Name4	*	128 Byte (string)
Parameter1	*	4 Byte (long int.)
Parameter2	*	4 Byte (long int.)
Parameter3	*	4 Byte (long int.)
Parameter4	*	4 Byte (long int.)

* zusätzlicher Name siehe Tabelle 5-23 Unterfunktionsnummer SFkt für die Allgemeine Auftragsfunktion

5.13.1 Laden von NC-Programmen

Funktionsaufruf

C_ORDER_M () mit
SFkt=2
Name1 = NC-Programmname
Name2 = NCK-Name (optional)

Parameter1 =	0:	Einzelnes NC-Program in NCK laden
	2:	komplettes Werkstück in NCK laden
	3:	Werkstück mit Ausnahme des in Name1 angegebenen NC-Programms in NCK laden

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 2, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 2, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,0 ,0 ,0);

Bedeutung

Laden von einzelnen NC-Programmen oder Werkstücken aus der SINUMERIK-Datenhaltung in die NCK. Der NCK-Name (Name2) muss nur dann angegeben werden, wenn das NC-Programm nicht in die Standard-NCK geladen werden soll.

Fehlernummer

-262: NC-Programm kann nicht geladen werden

5.13.2 Laden und Anwählen von NC-Programmen

Funktionsaufruf

C_ORDER_M () mit
SFkt=3
Name1 = Werkstück- / NC-Programmname
Name2 = NCK-Name (optional)

Parameter1 =	0:	NC-Programm in NCK laden und anwählen
	1:	Einzelnes Programm von Extern abarbeiten
	2:	komplettes Werkstück in NCK laden und NC-Programm aus Name1 anwählen
	3:	komplettes Werkstück in NCK laden und das in Name1 angegebene NC-Programm zur Abarbeitung von Extern anwählen.
Parameter2 =		Kanal-Nummer

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 3, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,2 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 3, "\WKS.DIR\Zylinderkopf.wpd\Kw15.mpf", "NCU_2", "\0", "\0",0 ,0 ,0 ,0);

Bedeutung

Die Funktion entspricht der SFkt2. Zusätzlich wird das angegebene Programm angewählt. Der NCK-Name (Name2) muss nur dann angegeben werden, wenn das NC-Programm nicht in die Standard-NCK geladen werden soll.

Hinweis

Soll das NC-Programm anschließend automatisch gestartet werden oder soll der Bediener eine Anzeige des NC-Programms erhalten, muss dies über den RPC SINUMERIK Dienst R_VAR_M realisiert werden.

Fehlernummer

-263: NC-Programm kann nicht geladen und angewählt werden

5.13.3 Entladen von NC-Programmen

Funktionsaufruf

C_ORDER_M () mit
SFkt=4
Name1 = NC-Programmname oder Werkstückname
Name2 = NCK-Name (optional)

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\WKS.DIR\Zylinderkopf.wpd\Kw15.mpf", "\0", "\0", "\0",0 ,0 ,0 ,0);

Bedeutung

Entladen von NC-Programmen oder Werkstücken aus der NCK. Anschließend befinden sich die NC-Programme in der PCU-Datenhaltung.
Der NCK-Name (Name2) muss nur dann angegeben werden, wenn nicht in die Standard-NCK geladen werden soll.

Fehlernummer

-264: NC-Programm kann nicht entladen werden

5.13.4 Anwählen von NC-Programmen

Funktionsaufruf

C_ORDER_M () mit
SFkt=5
Name1 = NC-Programmname
Name2 = NCK-Name (optional)

Parameter1 =	0:	auf NCK abarbeiten
	1:	von Extern abarbeiten
Parameter2 =		Kanal-Nummer

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 5, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",1 ,5 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 5, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,1 ,0 ,0);

Bedeutung

Anwählen von NC-Programmen in einem bestimmten Kanal. Die Funktion entspricht SFkt 3 **ohne** vorheriges Laden des NC-Programms.
Der NCK-Name (Name2) muss nur dann angegeben werden, wenn das NC-Programm nicht in den Standard-NCK geladen werden soll.

Fehlernummer

-265: NC-Programm kann nicht angewählt werden

5.13.5 Abwählen von NC-Programmen

Funktionsaufruf

C_ORDER_M () mit
SFkt=6
Name2 = NCK-Name (optional)
Parameter2 =Kanal-Nummer

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 6, "\0", "\0", "\0", "\0",0 ,5 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 6, "\0", "NCU_2", "\0", "\0",0 ,1 ,0 ,0);

Bedeutung

Abwählen des aktuellen NC-Programmes.
Der NCK-Name (Name2) muss nur dann angegeben werden, wenn es sich dabei nicht um die Standard-NCK handelt.

Fehlernummer

-266: NC-Programm kann nicht abgewählt werden

5.14 Datum/Uhrzeit auf SINUMERIK/PLC setzen

Funktionsaufruf

C_ORDER_M () mit
SFkt=100
Name2 = NCK-Name (optional)

Parameter1 =		UNIX-Zeit
Parameter2 =	0:	Datum/Uhrzeit auf SINUMERIK
	1:	Datum/Uhrzeit auf SINUMERIK und PLC
	2:	Datum/Uhrzeit auf PLC

Übertragungsrichtung **FLR → SINUMERIK**

Beispiel

C_ORDER_M ("FLR1", "BAZ3", 0, 100, "\0", "\0", "\0", "\0",862826400 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 100, "\0", "NCU_2", "\0", "\0",862826400 ,1 ,0 ,0);

Bedeutung

Datum und Uhrzeit auf SINUMERIK / PLC setzen.
Der NCK-Name (Name2) muss nur dann angegeben werden, wenn das Datum bzw. die Uhrzeit nicht auf der Standard-NCK geändert werden soll. Der Parameter2 weist das Zielgerät der Datums- bzw. Uhrzeitsänderung aus.

Fehlernummer

-270: Uhrzeit/Datum kann nicht aktualisiert werden

5.15 MODE-Umschaltung

Funktionsaufruf

C_MODE_M (Host,
Machine,
OrderNum,
Mode)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Kommando an die Maschine, eine bestimmte Betriebsart einzunehmen.

Daten

Tabelle 5-18 Parameter Mode-Umschaltung

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int.)
Mode	Betriebsart 1: Sondermodus ein 2: Sondermodus aus 3: Komponenten abschalten 4: Komponenten einschalten 5: in Anforderung von RPC SINUMERIK Bit 4 = 1 setzen 6: in Anforderung von RPC SINUMERIK Bit 4 = 0 setzen 7: in Anforderung von RPC SINUMERIK Bit 5 = 1 setzen 8: in Anforderung von RPC SINUMERIK Bit 5 = 0 setzen	4 Byte (long int.)

Mode 5 bis 8 kann zum projektspezifischen Ein- und Ausschalten von Sonderfunktionen benutzt werden.

Beispiel

C_MODE_M ("FLR1", "BAZ3", 0, 3);

5.15.1 Sondermodus

Beschreibung

Im Sondermodus werden durch den automatischen Materialfluss keine Werkstückträger an die Maschine angeliefert, auf die Werkstückträger-ankunftsmeldung erfolgt keine Programmzuordnung durch den FLR. Mit Handtransporten können Werkstückträger angeliefert werden, der Bediener muss selbst das NC-Programm anwählen und starten.
Werkstückträger, die über den automatischen Materialfluss angeliefert wurden, werden auch im Sondermodus automatisch weitertransportiert. Werkstückträger, die mit Handtransport angeliefert wurden, müssen auch mit Handtransport abtransportiert werden.
Am FLR darf der Sondermodus für eine Maschine zu einem beliebigen Zeitpunkt eingeschaltet werden, eine laufende Bearbeitung auf der Maschine läuft ganz normal zu Ende.
Sobald der Sondermodus eingeschaltet ist, darf mit Handtransport der im Sondermodus zu bearbeitende Werkstückträger angeliefert werden, die Maschine muss **nicht** leergefahren werden.
Mode = 1 Sondermodus einschalten
Mode = 2 Sondermodus abschalten

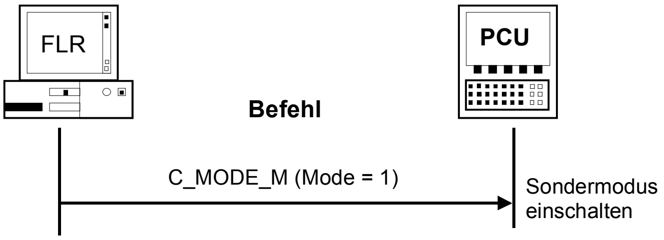


Bild 5-21 Sondermodus einschalten, Initiative durch FLR

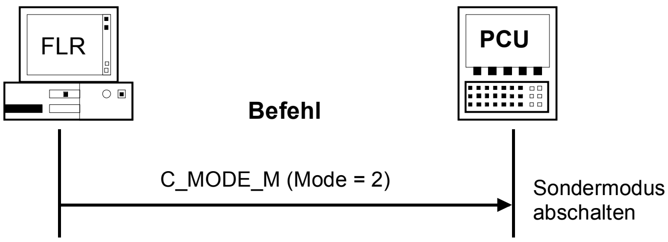


Bild 5-22 Sondermodus abschalten, Initiative durch den FLR

5.15.2 Komponenten abschalten/ einschalten

Um nach Arbeitsende Antriebe oder andere Anlagenkomponenten durch den FLR abschalten zu können, wird eine entsprechende Anforderung benötigt. Ebenso wird für den Arbeitsbeginn eine Einschalt-Anforderung benötigt.
Wann der FLR die Abschaltung anfordert, zu einer bestimmten Uhrzeit oder erst, nachdem der letzte Werkstückträger bearbeitet ist, spielt für die Rechnerkopplung keine Rolle. Sie gibt die Anforderung nur über die DB-Nahtstelle an die PLC weiter. Die PLC muss prüfen, ob keine Bearbeitung läuft, bevor sie die Komponenten abschaltet.
Nach Abschalten der Komponenten muss dies mit **R_MACHINE_H ()** dem FLR gemeldet werden, ebenso nachdem sie wieder eingeschaltet sind.

Mode = 3 Komponenten abschalten
Mode = 4 Komponenten einschalten

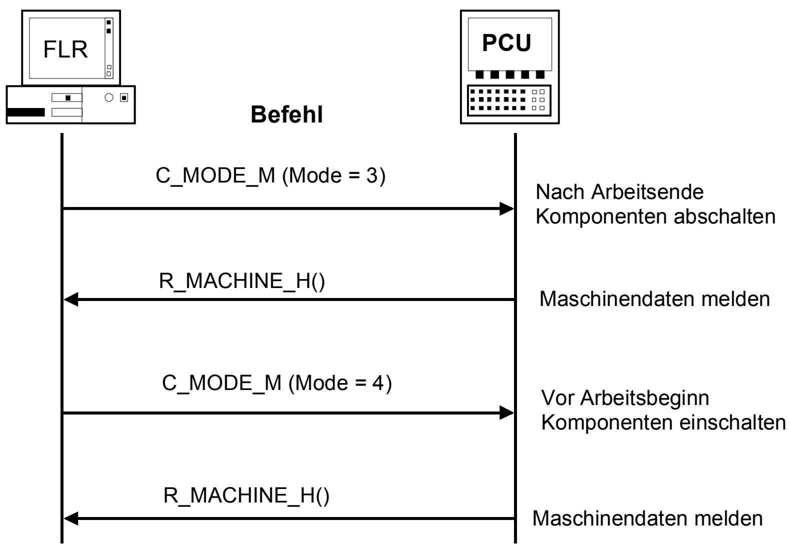


Bild 5-23 Feierabend, Initiative durch den FLR

5.16 Synchronisation

Synchronisation bedeutet, den FLR mit aktuellen Daten zu versorgen, damit das Systemabbild im Rechner mit der Realität übereinstimmt. Dies ist bei einem Wiederanlauf des Rechners, oder der Maschine, oder nach einer Verbindungsunterbrechung notwendig.
Während der Synchronisation darf die Maschine keine neue Bearbeitung beginnen, die laufende Bearbeitung wird von der Synchronisation nicht beeinflusst. Der FLR fordert von der Maschine mit **T_MACHINE_M ()** Maschinenzustandsdaten und mit **T_DATA_M (SFkt = 50)** Maschinenbelegungsdaten an, und überträgt für die noch nicht fertig bearbeiteten Werkstückträger die Programmzuordnung **R_NC4WPC_M ()** für alle Seiten.

Ist der FLR auch für Werkzeugbereitstellung / -Bilanzierung zuständig, sollte nach einer (längeren) Verbindungsunterbrechung im Anschluß an die Synchronisation der Maschine, eine Abfrage aller Werkzeuge erfolgen, damit der FLR wieder über die aktuellen Werkzeugdaten verfügt. Die WZ-Abfrage erfolgt nicht automatisch, der FLR muss sie bei Bedarf mittels
T_DATA_M (SFkt =20) anfordern.

5.16.1 Synchronisation- Start / -Ende

Funktionsaufruf

```
C_SYNCH_M (    Host,
               Machine,
               OrderNum,
               SynchFlag)
```

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Der FLR leitet einen Synchronisationsvorgang ein bzw. beendet ihn.

Daten

Tabelle 5-19 Beschreibung der Synchronisationsparameter

Parameter	**Beschreibung**	**Format**
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
SynchFlag	Start- / Endekennung 1: Start 0: Ende	4 Byte (long int)

Anwenderhinweise

Der folgende Abschnitt erläutert die Abläufe der Synchronisation im Zusammenwirken.

Beispiel

```
C_SYNCH_M ("FLR1", "BAZ3", 0, 1);
```

5.16.2 Ablauf der Synchronisation

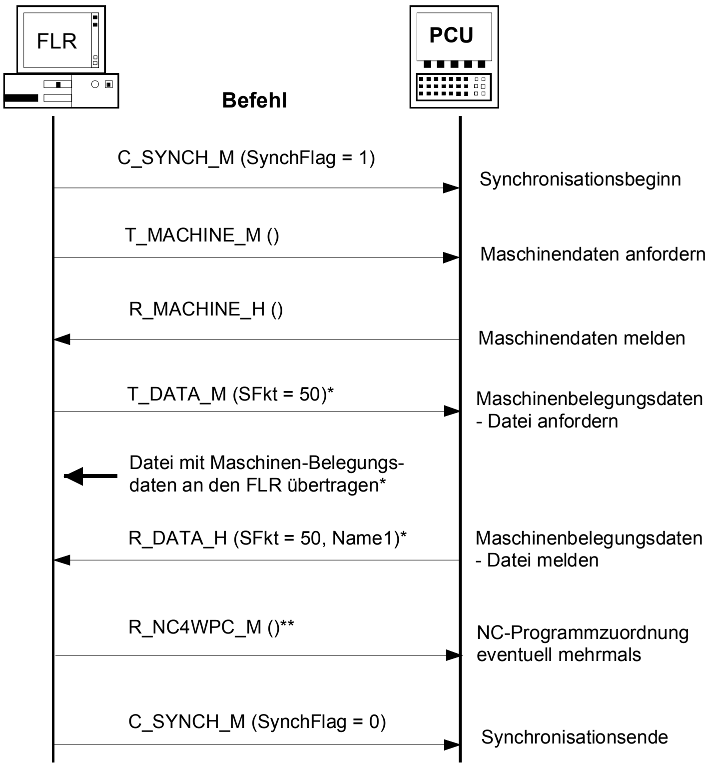
FLR → SINUMERIK	C_SYNCH_M ()	SynchFlag =1 //Start
FLR → SINUMERIK	T_MACHINE_M ()	
FLR ← SINUMERIK	R_MACHINE_H ()	

Bei Maschinen mit mehr als 3 Liegeplätzen, z.B. Transportsystem folgt noch:

FLR → SINUMERIK	T_DATA_M () SFkt = 50
FLR ← SINUMERIK	Maschinenbelegungsdaten-Datei übertragen
FLR ← SINUMERIK	R_DATA_H () SFkt = 50, Name1 = Dateiname

Für alle noch nicht fertig bearbeitete Werkstückträger und alle deren Seiten folgt je einmal

FLR → SINUMERIK	Programmzuordnung	R_NC4WPC_M ()
FLR → SINUMERIK	C_SYNCH_M ()	SynchFlag = 0 //Ende



Schritte * : Bei Maschinen mit mehr als 3 Liegeplätzen (z.B. Transportsystem)
Schritte** : Für alle noch nicht fertig bearbeitete Werkstückträger und alle deren Seiten je einmal

Bild 5-24 Synchronisation, Initiative durch den FLR

■

Für Notizen

6

6 Datenübertragung für OEM-Applikation

Um Daten von beliebigen OEM-Applikationen in der SINUMERIK über die Rechnerkopplung an den FLR senden zu können, sowie auch umgekehrt Daten vom FLR über die Rechnerkopplung an beliebige OEM-Applikationen, ist eine Kommunikation zwischen RPC SINUMERIK in der SINUMERIK und anderen Applikationen möglich.
Da die Kommunikation zwischen Programmen in der SINUMERIK über DDE erfolgt, wird auch für die Kommunikation zwischen Rechnerkopplung und OEM Applikationen DDE eingesetzt.
Es werden Funktionen und Schnittstellen für die Übertragung von Daten in beide Richtungen definiert.

6.1 Daten an eine OEM-Applikation

Funktionsaufruf

R_DDEDATA_M (Host,
Machine,
OrderNum,
Application,
Topic,
Item,
Data**)**

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Dieser RPC veranlasst die Übergabe der Nutzdaten mittels DDE an die angegebene OEM-Applikation. Alle dafür notwendigen Parameter werden bei R_DDEDATA_M übergeben.
Da es nicht möglich ist, während des RPC eine DDE Verbindung aufzubauen, kann die Datenübergabe an die OEM Applikation erst nach dem 'return' von R_DDEDATA_M erfolgen.
Für jeden R_DDEDATA_M () wird eine DDE Verbindung aufgebaut. Vor die Nutzdaten kopiert RPC SINUMERIK den Hostnamen gefolgt vom 'Pipe'-Symbol ('|' Zeichencode 124), um der OEM-Applikation den Absender mitzuteilen. Dieser gesamte String wird mit DDE-POKE an die OEM-Applikation übergeben, danach wird die DDE-Verbindung wieder abgebaut.
Als positive Quittung für den Absender von R_DDEDATA_M wird, nach Übergabe der Daten an die OEM Applikation, von RPC SINUMERIK R_REPORT_H mit Typ=5 und Fehlernummer = 1000 gesendet.

Hinweis

Es dürfen nur ASCII-Daten als Nutzdaten übergeben werden, keine Binärdaten. Der String endet mit binär 0.

Daten

Tabelle 6-1 Parameterbeschreibung Daten an FLR

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
Application	Applikationsname	64 Byte (string)
Topic	DDE Topic	64 Byte (string)
Item	DDE Item	64 Byte (string)
Data	DDE Nutzdaten	32 KB (string)

6.1.1 DDE zwischen einer OEM Applikation und RPC SINUMERIK

Damit eine OEM Applikation Daten mittels DDE an RPC SINUMERIK in der SINUMERIK übergeben kann, ist RPC SINUMERIK auch DDE-Server.
Die OEM Applikation muss dazu mit folgenden DDE-Paramtern eine DDE-Verbindung aufbauen und die Daten mittels POKE übergeben:

Application = Sincom*,
Topic = OEM,
Item = SendData.

Hinweis

Aus Kompatibilitätsgründen heißt der Applikationsname weiterhin Sincom.

Der beim DDE-Call übergebene Datenstring muss vor den eigentlichen Nutzdaten den Hostnamen enthalten, an den die Daten weiterzuleiten sind. Als Separator ist das ‘Pipe’-Symbol (‘|’ Zeichen-Code 124) zu verwenden (Hostnamen|Nutzdaten). Der Nutzdatenstring wird nach Empfang mit dem obigen R_DDEDATA_H () - Call an den angegebenen Host (FLR) weitergeleitet. Der Hostname muss der Rechnerkopplung bekannt sein (siehe Kapitel 5).

6.1.2 Dateiübertragung von OEM-Applikationen zum Host/FLR

Um von der OEM Applikation aus Dateien über die Rechnerkopplung an den FLR übertragen zu können, wird ein DDE-Call benutzt, der die bereits vorhandene RPC-Funktion R_DATA_H nutzt. Applikationsname und Topic sind gleich wie bei obigem DDE-Call, als Item ist File anzugeben:
Application = Sincom,
Topic = OEM,
Item = PutFile.
Der Datenstring muss die für R_DATA_H benötigten Paramter enthalten, als Separator ist das ‘Pipe’-Symbol (‘|’ Zeichen-Code 124) zu verwenden.
Aufbau des Datenstrings: Host|SFkt|Name1|Name2|Date|LastFile.
SFkt muss eine Nummer größer 1000 sein, sie dient zur Zordnung der Datei im FLR.
Name1 ist der Dateiname in der SINUMERIK (vollständig mit Laufwerk und Pfad)
Name2 ist der Dateiname im FLR (ebenfalls vollständig)
Date ist optional, es enthält Datum/Uhrzeit der Datei als Unix-Uhrzeit (in Sekunden).
LastFile ist optional, siehe Kapitel 5.

6.1.3 Dateiübertragung vom Host/FLR zur OEM-Applikation

Die Übertragung der Datei vom FLR zur SINUMERIK erfolgt mittels R_DATA_M (SFkt = 90). Bei SFkt 90 wird die Datei nur auf die SINUMERIK geholt, es findet keine weitere Bearbeitung der Datei statt. Um die OEM Applikation über die erhaltene Datei zu verständigen kann vom FLR mittels R_DDEDATA_M () eine entsprechende Nachricht übertragen werden.
Für OEM Applikationen muss die Übertragung von Dateien nur dann über die Rechnerkopplung abgewickelt werden, wenn die Übertragung über FTP erfolgen muss.
Kann die SINUMERIK über das Netz direkt auf die Laufwerke des FLR zugreifen, kann die OEM Applikation ihre Dateien selbst vom und zum FLR kopieren.

■

7

7 Projektierbare Datenübertragung/Variablen-Dienst

7.1 Beschreibung

Über die Variablendienste kann RPC SINUMERIK beliebige projektspezifische Daten aus PLC und NCK lesen und schreiben. Die Realisierung basiert auf dem Variablendienst des OEM Paketes PCU unter Nutzung des NC-DDE-Servers.
Der FLR kann über **T_VAR_M ()** Daten anfordern, die dann von RPC SINUMERIK ermittelt und über **R_VAR_H ()** dem FLR übergeben werden. Ebenso kann der FLR über **R_VAR_M ()** Daten übergeben, die dann von RPC SINUMERIK geschrieben werden.
In der Ini-Datei SCVARSET.INI im Verzeichnis ADD_ON können Variablen-Sets definiert werden, die für Lese- und Schreibvorgänge benutzt werden können. Bei entsprechend konfiguriertem Mode werden bei Änderungen in diesen Variablen, diese von RPC SINUMERIK automatisch mit einem **R_VAR_H ()** an den FLR gemeldet.
Für die SCVARSET.INI gelten folgende Festlegungen:
maximal können 10 Sets definiert werden
maximal können 10 Variablen in einem Set definiert werden
maximal können 50 Variablen innerhalb aller Sets definiert werden
maximal können 10 Hotlinks innerhalb aller Sets definiert werden
Unter Beachtung dieser Grenzen kann bedarfsgerecht kombiniert werden.

RPC SINUMERIK enthält die optionale Erweiterung dass im Variablenset, beim Anschluss von mehreren NCU's an eine SINUMERIK, die gewünschte NCU parametriert werden. Dies gilt jedoch nur bei Variablen-Sets die in der SCVARSET.INI definiert sind, nicht wenn die Datenbeschreibung direkt bei **T_VAR_M ()** oder **R_VAR_M ()** im Parameter VarDescr angegeben wird!

7.1.1 Aufbau der SCVASRSET.INI-Datei

Jedes Variablenset ist eine Sektion in der Inidatei und besteht aus dem Sektionsnamen in eckigen Klammern, dem Zugriffs-Mode, optional einem Hostnamen, optional einer NCU-Verbindung und den Variablendefinitionen.
Wird kein Hostname konfiguriert, wird **R_VAR_H ()** bei Datenänderungen an alle für diesen RPC projektierten Hosts gesendet.
Wird keine NCU-Verbindung konfiguriert wird die Default-Verbindung benutzt.
Der Name des Variablensets und der Hostname dürfen maximal 16 stellig sein, NCU-Verbindungen müssen mit den Konfigurationen in der Datei NETNAMES.INI übereinstimmen.

Zugriff-Modi

0	Es wird kein Hotlink eingerichtet, die Variablen des Sets werden nur auf Anforderung des FLR gelesen (T_VAR_M).
1*	Es wird für jede Variable im Set ein Hotlink eingerichtet,
2*	Es wird nur ein HotLink auf die erste Variable des Sets eingerichtet
3*	Mit Handshake. Wie bei 2 wird nur ein HotLink auf die erste Variabledes Sets eingerichtet. Nach einem Hotlink wird nach dem Lesen aller Variablen, die erste Variable mit 0 abgelöscht (Handshake). Die Variable 1 darf hierfür kein Feld definieren sondern nur eine einzelne Variable. Die Variable darf nach dem Löschen mit 0 erst nach einer Zeitverzögerung wieder gesetzt werden. Bei zu frühem erneuten Setzen der Variablen könnte es passieren, dass die Zustandsänderung nicht erkannt wird.

* Bei einem Hotlink werden alle Variablen des entsprechenden Variablensets gelesen und mit R_VAR_H an den FLR gesendet.

SCVASRSET.INI für RPC SINUMERIK-Variablendienst

```
; Name des Variablen-Sets
[MessWerte]
;Zugriffs-Mode
Mode=2
;Hostname (optional)
Host=H1
;NCU-Verbindung (optional)
Connection=NCU_1
VAR01=/Plc/DataBlock/Word[c50,0]
VAR02=/Plc/Datablock/Byte[c50,1,20]("!I%d,")
VAR03=/Channel/Parameter/R[5]

[Set02]
Mode=3
Host=FLR2
Var01=/Plc/Datablock/Byte[c50,1]
Var02=/Plc/Datablock/Byte[c50,2]
Var03=/Plc/Datablock/Byte[c50,3,20]("!I%d,")
Var04=/Plc/Datablock/Byte[c50,4]
Var05=/Plc/Datablock/Byte[c50,5,20]("!I%d,")
Var06=/Plc/Datablock/Byte[c50,6]
Var07=/Plc/Datablock/Byte[c50,7,20]("!I%d,")
Var08=/Plc/Datablock/Byte[c50,8]
Var09=/Plc/Datablock/Byte[c50,9,20]("!I%d,")
Var10=/Plc/Datablock/Byte[c50,10]

[Set03]
Mode=0
Var01=/Plc/Datablock/Byte[c51,0,10]("!I%d,")
Var02=/Plc/Datablock/Byte[c51,30,50]("!I%d,")
```

Hinweis

Die SINUMERIK Software enthält ein Testtool mit dem Namen DDETEST. EXE. Damit kann bei laufendem NCDDE Server der Variablenzugriff getestet werden. Sie sollten alle Variablen, die Sie im SCVARSET.INI benutzen wollen, zuerst auf diesem Weg testen.
Im Anhang ist ein Kapitel der OEM Beschreibung enthalten, das die Beschreibung der Schnittstelle zwischen PCU50 und NCK/PLC beinhaltet
Das SCTEST Setup enthält die Hilfedatei Btss_gr.hlp (und Btss_uk.hlp für Englisch), Diese Hilfe-Dateien beinhalten alle Informationen für den Variablenzugriff.
Der Zugriff auf Daten über $-Variablen, z.B. $TC_TP1 (T-Nummer) , wird mit diesem Dienst nicht realisiert.

7.2 Daten übertragen

7.2.1 Variable Datenübertragung an die Maschine

Funktionsaufruf

R_VAR_M (Host,
Machine,
OrderNum,
VarMode
VarSet
VarDescr
VarData)

Übertragungsrichtung: FLR → SINUMERIK

Bedeutung

Daten vom FLR an RPC SINUMERIK übergeben, die dann in PLC oder NCK geschrieben werden.
M Aufruf R_VAR_M kann entweder bei VarSet der Name eines Variablensets aus der SCVARSET.INI angegeben werden, oder bei VarDescr werden die Variablenbeschreibungen angegeben.
Als positive Quittung für den Absender von R_VAR_M wird, nach Schreiben der Daten von RPC SINUMERIK R_REPORT_H mit Typ=5 und Fehlernummer = 0 gesendet.

Daten

Tabelle 7-1 Variable Datenübertragung an die Maschine

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
VarMode	Variablen-Modus	4 Byte (long int)
VarSet	Name des Variablen-Sets	16 Byte (string)
VarDescr *	Datenbeschreibung	1024 Byte (string)
VarData *	Nutzdaten	10 kB (string)

Sowohl bei der Datenbeschreibung als auch bei den Nutzdaten sind die einzelnen Variablen durch ‘Pipe’ Symbol (‘|’ Zeichen-Code 124) als Separator getrennt.

Beispiel 1

in SCVARSET.INI ist das SET03 definert:
[Set03]
Mode=1
Host=FLR2
Var01=/Plc/Datablock/Byte[c51,0,10]("!I%d,")
Var01=/Plc/Datablock/Byte[c51,30,50]("!I%d,")
R_VAR_M ("FLR1", "BAZ3", 0, 0, "Set03", "\0", "33|50");

Beispiel 2

R_VAR_M ("FLR1", "BAZ3", 0, 0, "\0",
"/Plc/DataBlock/Word[c50,20]|/Plc/DataBlock/Word[c60,30]|", "33|50");

7.2.2 Variable Datenübertragung an den FLR

Aufgerufene Funktion

R_VAR_H (Host,
Machine,
OrderNum,
VarMode
VarSet
VarDescr
VarData**)**

Übertragungsrichtung: **FLR ← SINUMERIK**

Bedeutung

RPC SINUMERIK übergibt aus PLC oder NCK gelesene Daten an den FLR.

Daten

Tabelle 7-2 Variable Datenübertragung an den FLR

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
VarMode	Variablen-Modus	4 Byte (long int)
VarSet	Name des Variablen-Sets	16 Byte (string)
VarDescr *	Datenbeschreibung	1024 Byte (string)
VarData *	Nutzdaten	10 KB (string)

*Sowohl bei der Datenbeschreibung als auch bei den Nutzdaten sind die einzelnen Variablen durch ‘Pipe’ Symbol (‘|’ Zeichen-Code 124) als Separator getrennt.
Es werden Variablensets gemeldet, die in der SCVARSET.INI definert sind.

Beispiel:

```
R_VAR_H ("FLR1", "BAZ3", 0, 0, "Set02", "\0", "33|50");
```

7.3 Daten anfordern

7.3.1 Variable Daten von der Maschine anfordern

Funktionsaufruf

T_VAR_M (Host,
Machine,
OrderNum,
VarMode
VarSet
VarDescr)

Übertragungsrichtung: FLR → SINUMERIK

Bedeutung

Der FLR fordert RPC SINUMERIK auf, Daten aus der PLC oder dem NCK zu lesen. Die Daten werden dann mit R_VAR_H () zurückgeliefert. Bei VarSet darf nur ein Variablenset angegeben werden, das in der SCVARSET.INI definert ist. VarDescr wird nicht ausgewertet.

Daten

Tabelle 7-3 Variable Daten von der Maschine anfordern

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
VarMode	Variablen-Modus	4 Byte (long int)
VarSet	Name des Variablen-Sets	16 Byte (string)
VarDescr *	Datenbeschreibung	1024 Byte (string)

*Sowohl bei der Datenbeschreibung als auch bei den Nutzdaten sind die einzelnen Variablen durch ‘Pipe’ Symbol (‘|’ Zeichen-Code 124) als Separator getrennt.

Beispiel

T_VAR_M ("FLR1", "BAZ3", 0, 0, "Set02", "\0", "\0");

7.3.2 Variable Daten vom FLR anfordern

Aufgerufene Funktion

T_VAR_H (Host,
Machine,
OrderNum,
VarMode
VarSet
VarDescr
VarData)

Übertragungsrichtung: FLR ← SINUMERIK

Bedeutung

RPC SINUMERIK fordert vom FLR Daten an. Die Daten werden dann mit R_VAR_M () an RPC SINUMERIK geliefert.

Hinweis

Dieser RPC wird zur Zeit von RPC SINUMERIK nicht benutzt.

■

Für Notizen

8

8 Kommunikation zwischen FLR und TPS

8.1 Schnittstelle TPS/Maschine

Das Transportsystem (TPS) erhält Transportaufträge vom FLR und muss diese auf Ausführbarkeit (logisch und physikalisch) prüfen. Dazu muss er prüfen, ob das Ziel frei und nicht gesperrt ist.

Haltestellen

Die Übergabepunkte vom TPS zur Bearbeitungsmaschine werden als Haltestellen (docking places) bezeichnet.
Es gibt 3 Arten von Haltestellen:
Bringe-Haltestellen (Input), es kann nur angeliefert werden
Hole-Haltestellen (Output), es kann nur abgeholt werden
Bringe-/Hole-Haltestellen (Input/Output), es kann angeliefert und abgeholt werden

8.2 TPS-Zustandsdaten

Aufgerufene Funktion

```
R_TPS_H (    Host,
             Machine,
             OrderNum,
             Mode,
             MachineStatus,
             TpoStatus,
             DockPos,
             DockPosStatus,
             WPC,
             Resint1,
             Resint2,
             Resbyte)
```

Übertragungsrichtung: **FLR ← SINUMERIK**

Bedeutung

TPS-Zustandsdaten zum Fertigungsleitrechner übertragen.

Daten

Tabelle 8-1 Parameter der Transportsystem-Zustandsdaten

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	TPS-Name	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
Mode	Betriebsart SINUMERIK-Betriebsart: 1: Automatik 2: MDA 4: JOG Rechnerkopplung-Betriebsart: 100: FLR-Modus 300: Manueller-Modus 1000: bei Bit 6=1 im RPC SINUMERIK-Modus 2000: bei Bit 7=1 im RPC SINUMERIK-Modus	4 Byte (long int)
MachineStatus	Maschinenzustand 0: Neuanlauf 1: untätig 2: ätig 3: gestört 4: Feierabend (Antriebe abgeschaltet)	4 Byte (long int)
TpOStatus	Transportauftragszustand 0: Keine Transportauftragsdaten 1: Neuer Transportauftrag (nicht begonnen) 2: in Bearbeitung 4: WPC ist auf dem Wagen 8: Auftrag abgeschlossen 16: Fehler, Auftrag nicht ausführbar 32: Fehler, Ersatzziel angefahren (DockPos)	4 Byte (long int)
DockPos[2]	Haltestellennummer Die Haltestellennummer entspricht dem Index in der Haltestellenliste des Nahtstellen DBs beginnend bei 1. Haltestellennummer = 0 bedeutet: „nicht belegt“.	4 Byte (long int)
DockPosStatus[2]	Haltestellenzustand 0: freigegeben 1: für TPS gesperrt 2: gestört	4 Byte (long int)
WPC[2]	Werkstückträgerbezeichnung	6 Byte (string)*
Resint1 **	Reserve 1	4 Byte (long int)
Resint2 **	Reserve 2	4 Byte (long int)
Resbyte	Reserve 3	8 Byte (string)

* WPC ist als 2-dimensionales char-Array (char [2][6])definiert, jede der 2 Werkstückträgerbezeichnungen muss jedoch als String, abgeschlossen mit ‘\0’ angegeben werden.

** Resint1 und Resint2 sind an der DB-Nahtstelle der PLC aufgelegt. Werden von der PLC dort Werte eingetragen, so werden sie an den FLR übertragen. Diese Werte haben für die Rechnerkopplung keine Bedeutung. Sie werden an den FLR durchgereicht.

Anwendungshinweise

Das TPS setzt diesen RPC bei jeder Zustandsänderung, Betriebsartenumschaltungen und bei Beginn und nach Ende von Transporten ab. Beim Start von RPC SINUMERIK in der SINUMERIK wird an jeden projektierten FLR dieser RPC als Anlaufmeldung gesendet.

8.3 TPS-Zustandsdaten anfordern

Funktionsaufruf

T_TPS_M (Host,
Machine,
OrderNum)

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Transportsystemzustandsdaten von der SINUMERIK anfordern.

Daten

Tabelle 8-2 Parameter der TPS-Zustandsdatenanforderung

Parameter	**Beschreibung**	**Format**
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	TPS-Name	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)

Anwendungshinweis

Der FLR kann mit diesem Call die TPS-Zustandsdaten anfordern, dies ist z.B. bei der Synchronisation erforderlich. Das TPS meldet dann die Daten mit **R_TPS_H()**.

Beispiel

T_TPS_M ("FLR1", "BAZ3", 0);

8.4 Transportauftrag

Funktionsaufruf

```
C_TPORDER_M ( Host,
              Machine,
              OrderNum,
              SDockPos,
              DDockPos,
              WPC,
              WPCTyp,
              BufferFlag,
              Priority,
              ChainNum,
              Vehicle,
              Resint1,
              Resint2,
              Resbyte)
```

Übertragungsrichtung: **FLR → SINUMERIK**

Bedeutung

Transportauftrag zur Maschine senden.

Daten

Tabelle 8-3 Parameter des Transportauftrages

Parameter	Beschreibung	Format
Host	Fertigungsleitrechnername (Host)	16 Byte (string)
Machine	Maschinenname	16 Byte (string)
OrderNum	Auftragsnummer	4 Byte (long int)
SDockPos	Quell-Haltestellennummer Die Haltestellennummern entsprechen dem Index in der Haltestellenliste des Nahtstellenbausteins beginnend bei 1.	4 Byte (long int)
DDockPos	Ziel-Haltestellennummer Die Haltestellennummern entsprechen dem Index in der Haltestellenliste des Nahtstellenbausteins beginnend bei 1.	4 Byte (long int)
WPC	Werkstückträgerbezeichnung	6 Byte (string)
WPCTyp	Werkstückträgertyp	4 Byte (long int)
BufferFlag	Ziel dient nur zur Pufferung	4 Byte (long int)
Priority	Transportpriorität	4 Byte (long int)
ChainNum	Auftragskettungsnummer	4 Byte (long int)
Vehicle	Transportwagennummer	4 Byte (long int)
Resint1	Reserve 1	4 Byte (long int)
Resint2	Reserve 2	4 Byte (long int)
Resbyte	Reserve 3	8 Byte (string)

Beispiel

C_TPORDER_M ("FLR1", "BAZ3", 0, 3, 4, "WPC05", 0, 0, 0, 1, 0, 0, "\0");

Quell- und Ziel-Haltestelle

Quell- und Zielhaltestelle enthalten den Index auf den entsprechenden Haltestellendatensatz in der Haltestellenliste des Nahtstellen DBs, wobei der Index mit 1 beginnt.

Werkstückträger

Die Werkstückträgerbezeichnung kann vom Transportsystem zu Plausibilitätsprüfungen benutzt werden.
Der Werkstückträgertyp ist eine Zusatzinformation für das Transportsystem, sie kann die Art oder die Größe eines Werkstückträgers enthalten.
Mit dem Parameter BufferFlag = 1 kann der FLR dem Transportsystem mitteilen, wenn ein Werkstückträger nur zur Pufferung auf eine Maschine zu transportieren ist, er aber dort nicht bearbeitet werden soll (Hilfspufferplatz). Diese Information muss eventuell vom TPS an die Maschine übergeben werden. Bei allen anderen Transporten muss BufferFlag = 0 gesetzt werden.

Transportpriorität

Die Transportpriorität ist eine Zusatzinformation, falls dem Transportsystem mehrere Aufträge übergeben werden dürfen. Durch die Priorität könnte dann die Reihenfolge ihrer Abarbeitung beinflusst werden.

Auftragskettung

Die Auftragskettung ist eine Zusatzinformation . Bei Transportwagen mit zwei Liegeplätzen und Maschinen mit nur einer Haltestelle können zwei Transportaufträge durch eine gemeinsame Auftragskettungsnummer logisch verknüpft werden.

z.B. Der erste Transportauftrag enthält die Anlieferung eines unbearbeiteten Werkstückträgers an eine Maschine. Auf der Maschine liegt noch ein fertigbearbeiteter Werkstückträger, der zweite Transportauftrag mit gleicher Auftragskettungsnummer wie der erste beinhaltet das Abholen des bearbeiteten Werkstückträgers von der Maschine auf einen Puffer- oder Spannplatz. In diesem Fall wird die neue Palette geholt und zur Maschine gefahren, dort wird dann zuerst der bearbeitete Werkstückträger auf den freien Platz des Wagen umgespeichert, erst dann kann der erste Auftrag durch umspeichern des unbearbeiteten Werkstückträgers auf die Maschine abgeschlossen werden. Danach kann zum Ziel von Transport zwei gefahren werden und auch dieser Auftrag abgeschlossen werden.

Transportwagen

Transportwagen ist eine Zusatzinformation. Bei Transportsystemen mit mehreren Transportwagen kann vom FLR vorgegeben werden mit welchem Wagen ein Transport durchgeführt werden soll.

8.4.1 Ablauf eines Transportes

Übertragungs-richtung	Befehl	Bemerkung
FLR → SINUMERIK	C_TPORDER_M ()	
FLR ← SINUMERIK	R_TPS_H () (optional)	meldet TPS tätig, entleerten Liegeplatz und WPC auf dem Wagen
FLR ← SINUMERIK	R_TPS_H ()	meldet TPS untätig, neuen Liegeplatz des WPC und leeren Wagen

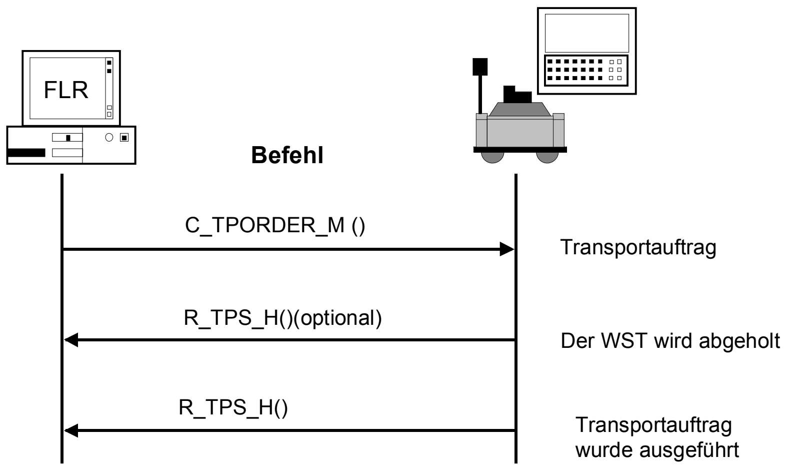


Bild 8-1 Transportdialog, Normalfall fehlerfrei

8.4.2 Fehler bei Transportaufträgen

Im Fehlerfall setzt das TPS eine Fehlermeldung an den FLR ab, mögliche Fehler bei Transportaufträgen sind:
In RPC SINUMERIK verwendete Fehlernummern:

Tabelle 8-4 Fehlernummern bei Transportsystem

Nr.	Bedeutung
-200	Transportauftrag kann nicht in den Nahtstellen DB geschrieben werden
-700	Transportauftrag wird vom Transportsystem mit Fehler gemeldet.

8.5 Synchronisation Transportsystem (TPS)

Die Synchronisation des TPS läuft ab wie bei der Maschine, der FLR fordert vom TPS Zustands- und Belegungsdaten an.
Während der Synchronisation darf das Transportsystem keine Zustandsänderungen melden.

Übertragungsrichtung	**Befehl**
FLR → SINUMERIK	C_SYNCH_M () SynchFlag = Start
FLR → SINUMERIK	T_TPS_M ()

Läuft während der Synchronisation schon ein Transportauftrag, darf dessen Ende erst nach der Synchronisation gemeldet werden.

Übertragungsrichtung	**Befehl**	**Bemerkung**
FLR ← SINUMERIK	R_TPS_H ()	
FLR → SINUMERIK	T_DATA_M () SFkt = 50	
FLR ← SINUMERIK		Maschinen-belegungsdaten-Datei an den FLR übertragen
FLR ← SINUMERIK	R_DATA_H () SFkt = 50, Name 1 = Dateiname	
FLR → SINUMERIK	C_SYNCH_M () SynchFlag = Ende	

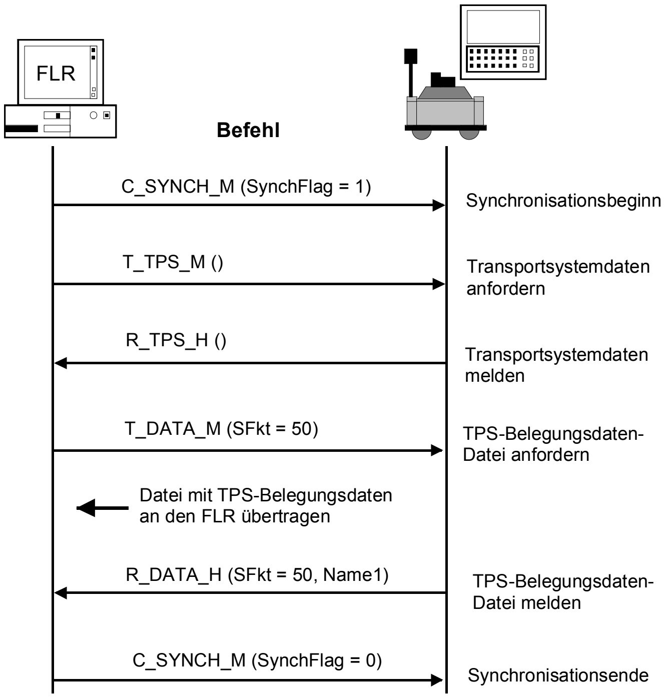


Bild 8-2 TPS-Synchronisation, Initiative durch FLR

■

Für Notizen

9

9 Zusammenfassung der RPC-Calls

9.1 Funktionsaufrufe von FLR an SINUMERIK

Tabelle 9-1 Funktionsaufrufe von FLR an SINUMERIK

Aufruf	Beschreibung
T_MACHINE_M ()	Anstoß von „Server“ an Client weiterleiten
T_TPS_M ()	Anstoß von „Server“ an Client weiterleiten
T_REPORT_M ()	Anstehende Alarme anfordern
R_NC4WPC_M ()	in Liste eintragen, Anstoß an Client
R_REPORT_M()	Meldung an Maschine übertragen
C_DELETE_M ()	Löschen von Daten in der Datenhaltung
C_MODE_M ()	Bits in der Nahtstelle setzen
C_SYNCH_M ()	Bit in der Nahtstelle setzen / rücksetzen
C_TPORDER_M ()	in Liste eintragen, Anstoß an Client
C_ORDER_M ()	Allgemeine Auftragsfunktion
T_DATA_M ()	
R_DATA_M ()	
T_VAR_M ()	
R_VAR_M ()	
R_DDEDATA_M ()	
R_MESSAGE_M ()	

9.2 Funktionsaufrufe von SINUMERIK an FLR

Tabelle 9-2 Funktionsaufrufe von SINUMERIK an FLR

Aufruf	Beschreibung
R_MACHINE_H ()	Maschinenzustandsdaten dem FLR anbieten
R_TPS_H ()	Transportsystem-Zustandsdaten dem FLR anbieten
R_REPORT_H ()	Meldungen an den FLR übergeben
T_DATA_H ()	
R_DATA_H ()	
T_VAR_H ()	Option
R_VAR_H ()	
R_DDEDATA_H ()	
R_MESSAGE_H ()	

Hinweis

Nicht alle Unterfunktionsnummern (SFkt) sind in beide Richtungen möglich.

Tabelle 9-3 Unterfunktionsnummer SFkt für Datendialoge

Unterfkt.-Nr.	**Funktion**	**Befehl**	**Bemerkung**
1	NC-Programm	T_DATA_H () T_DATA_M () R_DATA_H () R_DATA_M ()	Name1 = NC-Programm Name2 = Dateiname mit Pfad auf dem FLR
10	Liste der vorhandenen NC-Programme	T_DATA_M () R_DATA_H ()	
20	Werkzeugzustandsdaten aller Werkzeuge Variante 1, vollständiger WZ-Datensatz	T_DATA_M () R_DATA_H ()	Name1 = leer Name2 = Dateiname mit Pfad
21	Werkzeugzustandsdaten eines WZ Variante 2 reduzierter WZ-Datensatz	T_DATA_M () R_DATA_H ()	Name1 = Identnummer, Duplonr. Name2 = Dateiname mit Pfad
22	Werkzeugzustandsdaten eines WZ Variante 3 reduzierter WZ-Datensatz	T_DATA_M () R_DATA_H ()	Name1 = Identnummer, Duplonr. Name2 = Dateiname mit Pfad
23	Werkzeugzustandsdatendaten aller WZ Variante 3 reduzierter WZ-Datensatz	T_DATA_M () R_DATA_H ()	Name1 = Identnummer, Duplonr. Name2 = Dateiname mit Pfad
24	WZ-Daten eines WZ mit Adapternummer Vollständiger WZ-Datensatz	T_DATA_H R_DATA_M	
26	Wahlfreies-/manuelles Beladen eines WZ Vollständiger WZ-Datensatz, WZ-Daten 1	T_DATA_H () R_DATA_M ()	
27	Wahlfreies-/manuelles Entladen eines WZ	R_DATA_H	
28	WZ Beladen von WZ-Palette	T_DATA_H () R_DATA_M ()	
29	WZ Entladen auf WZ Palette	T_DATA_H () R_DATA_M ()	
50	Maschinenbelegungsdaten	T_DATA_M () R_DATA_H ()	
90	Beliebige Datei	T_DATA_M () R_DATA_H ()	

Tabelle 9-4 Unterfunktionsnummer SFkt für die Allgemeine Auftragsfunktion

Unterfkt.-Nr. (SFkt)	**Funktion**	**Bemerkung**	
2	NC-Programm laden	Name1 = Name2 = Parameter1=	NC-Programmname NCK-Name (optional) Funktion (0,2,3)
3	Laden und Anwählen von NC-Programmen	Name1 = Name2 = Parameter1= Parameter2=	NC-Programmname NCK-Name (optional) Funktion (0-3) Kanalnummer
4	Entladen von NC-Programmen	Name1 = Name2 =	NC-Programm NCK-Name
5	Anwählen von NC-Programmen	Name1 = Name2 = Parameter1= Parameter2=	NC-Programmname NCK-Name (optional) Funktion (0,1) Kanalnummer
6	Abwählen von NC-Programmen	Name2 = Parameter2=	NCK-Name (optional) Kanalnummer
100	Datum/Uhrzeit auf SINUMERIK/PLC setzen	Name2 = Parameter1 Parameter2=	NCK-Name (optional) UNIX-Zeit Datum / Uhrzeit SINUMERIK / PLC

■

10

10 RPC SINUMERIK-OCX

10.1 Einführung

Was ist RPC SINUMERIK-OCX?

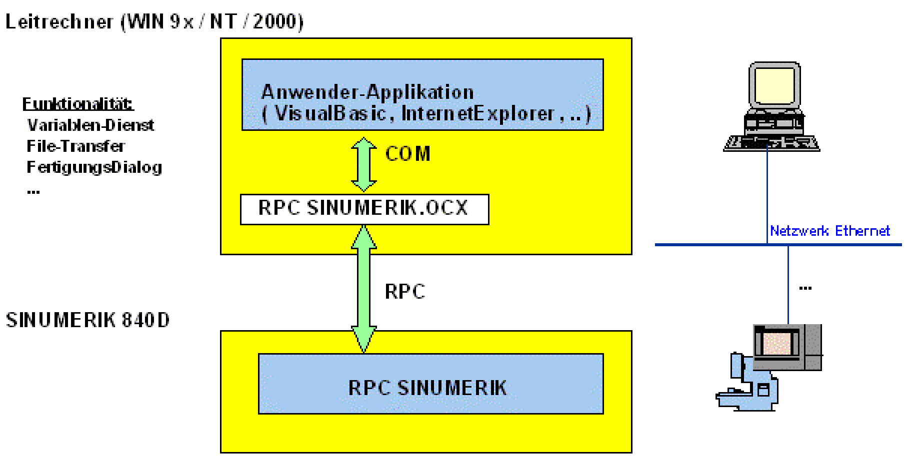
Das RPC SINUMERIK-OCX Entwicklungspaket stellt ein Zusatzprodukt zu der RPC SINUMERIK Rechnerkopplung dar.
Die RPC SINUMERIK Rechnerkopplung bietet eine Schnittstelle für die Kommunikation zwischen einer SINUMERIK 840D Maschinensteuerung und einem übergeordnetem Leitrechner. Die Kommunikation zwischen dem Leitrechner und der Maschinensteuerung erfolgt in dieser Konstellation über RPC's (Remote Procedure Call). Da die RPC's einen plattformunabhängigen Standard darstellen, ist die RPC SINUMERIK-Schnittstelle dadurch sowohl von MS-Windows als auch von UNIX, Linux und anderen Systemen anwendbar.

Einsatzbereich

Jedoch ist die Benutzung der RPC's meistens nur aus der Programmiersprache C/C++ möglich.
Unter Verwendung der RPC SINUMERIK-OCX wird die RPC SINUMERIK–Schnittstelle für eine ganze Reihe von verbreiteten Windows-Entwicklungssystemen zugänglich, ohne dass eine C/C++ Programmierung notwendig wird. Es werden alle Entwicklungssysteme unterstützt, die das Einbinden von 32Bit-ActiveX Komponenten beherrschen. Dazu zählen unter anderen : MS Visual Basic ab V4.0 (32Bit), MS Visual J++ 6.0, Internet Explorer ab V4.0, WinDev und viele andere Entwicklungssysteme.

Arbeitsweise

Die RPC SINUMERIK-OCX kapselt die RPC-Schnittstelle von RPC SINUMERIK in COM-Aufrufe (Component Object Model). Die Funktionsweise der einzelnen RPC's wird durch die Verwendung von RPC SINUMERIK-OCX nicht verändert. Die Funktionsweise der RPC's ist in der RPC SINUMERIK Dokumentation beschrieben.

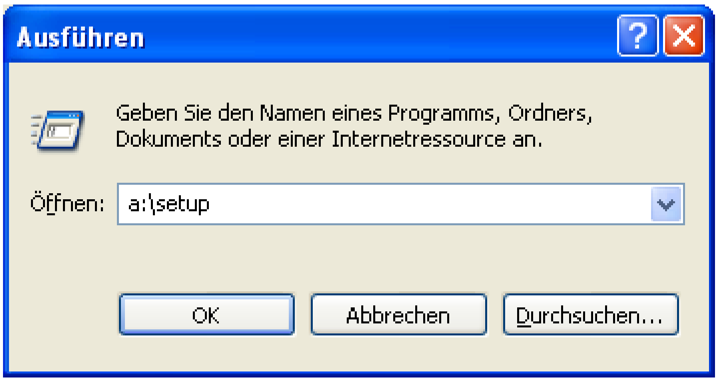


Betriebssysteme

Die RPC SINUMERIK-OCX ist anwendbar auf WIN 9x/NT/2000/XP/2003-Rechnern mit installiertem TCP/IP Netzwerk. Als Kommunikationspartner wird eine oder mehrere SINUMERIK 840D Steuerungen mit RPC SINUMERIK vorausgesetzt.

10.2 Installation des RPC SINUMERIK-OCX Entwicklungs-paketes

Die Installation des RPC SINUMERIK-OCX Entwicklungspaketes wird durch die Ausführung von Setup.exe aus der ersten Installations-Diskette durchgeführt.



Im Verlauf der Installation wird das Zielverzeichnis und der Name des Programmordners im Startmenü abgefragt.

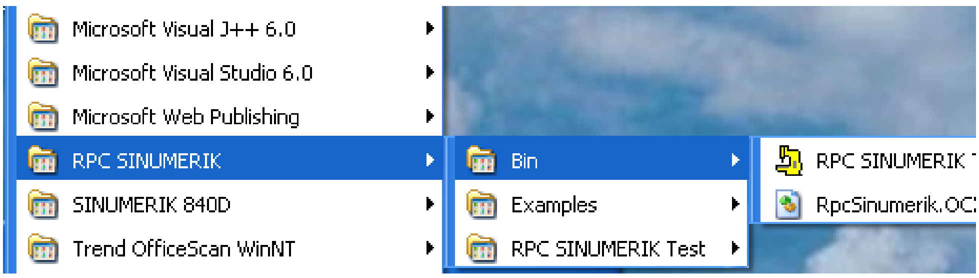
Es werden folgende Verzeichnisse unterhalb des Zielverzeichnisses angelegt:

Tabelle 10-1 Verzeichnisse unterhalb des Zielverzeichnisses

Doc	Verzeichnis mit dieser Dokumentation
Bin	Verzeichnis mit der RpcSinumerik.OCX und der Applikation RpcSinumerikTest.EXE
RPC SINUMERIK Test	Verzeichnis mit dem Quell-Code der Applikation RPC SINUMERIK Test
Examples	Verzeichnis mit Beispielen der Anwendung der RPC SINUMERIK-OCX

Darüber hinaus wird das Microsoft Visual Basic 6.0 (SP3) Laufzeitsystem im <Windows>\System - Verzeichnis installiert, falls noch nicht vorhanden.

Sie erreichen die installierten Dateien über das Start-Menü:



Test und Beispiele

Nach der Installation kann die RPC SINUMERIK Schnittstelle unter der Verwendung von RPC SINUMERIK-OCX mit der Applikation RPC SINUMERIK Test getestet werden. Voraussetzungen für den Einsatz der Applikation RPC SINUMERIK Test entnehmen Sie bitte dem Kapitel : „10.4 Testapplikation RPC SINUMERIK Test“. Nach einem erfolgreichem Test der Verbindung zur RPC SINUMERIK können die im Kapitel: „10.5 Beispiele der Anwendung der RPC SINUMERIK-OCX“ beschriebenen Beispiele verwendet werden.

10.3 Beschreibung der RPC SINUMERIK-OCX Komponente

Die RPC SINUMERIK-OCX ist als eine 32Bit ActiveX/COM Komponente realisiert. Sie kann in allen 32Bit Windows-Entwicklungssystemen eingesetzt werden, die das Einbinden von solchen Komponenten erlauben verwendet werden.

Pro Verbindung zur der Maschinen-Steuerung muss in der Anwender-Applikation eine separate Instanz der RPC SINUMERIK-OCX Komponente instanziert werden.

10.3.1 Installation

Mit der Installation des RPC SINUMERIK-OCX Entwicklungspaketes wird die Datei RpcSinumerik.OCX im <Windows>\System Verzeichnis hinterlegt und in der Windows© Registry registriert.

Weitere Rechner

Wenn Sie die RPC SINUMERIK-OCX Komponente auf weiteren Rechner verwendet wollen, muss sie dort ebenfalls installiert werden. Dies kann von Installationsprogrammen (wie z.B.: InstallShield) oder mit folgenden manuell Schritten vorgenommen werden.

1. Die Datei RpcSinumerik.OCX aus dem Verzeichnis C:\Programme\Siemens\MCIS\RPC SINUMERIK\bin kopieren in das <Windows>\System Verzeichnis des Rechners.
2. Die RPC SINUMERIK-OCX Komponente registrieren in der Windows-Registry mit folgenden Befehl:
   Regsvr32 <Windows>\System\RpcSinumerik.OCX

Darüber hinaus muss eine TCP/IP Verbindung zur RPC SINUMERIK Steuerung bestehen.

10.3.2 Attribute der RPC SINUMERIK-OCX Komponente

Die RPC SINUMERIK-OCX Komponente besitzt folgende Attribute, die zur Konfiguration der Verbindung eingesetzt werden.

Tabelle 10-2 Attribute der RPC SINUMERIK-OCX Komponente

Attribute	Bedeutung	Beispiel
MachineID	Name der Maschine in der RPC SINUMERIK-Konfiguration. Frei wählbare Bezeichnung der Maschine. Diese Information wird in jedem RPC übergeben zur Identifizierung der Kommunikationspartner.	M1
MachineIP	IP-Netzwerkadresse der Maschinen-Steuerung. Diese Information kann der Windows-Netzwerkinstallation entnommen werden. Es wird eine feste IP Vergabe vorausgesetzt. In der RPC SINUMERIK-OCX kann an dieser Stelle alternativ der Netzwerkname des jeweiligen Rechners verwendet werden.	195.2.208.233
MachinePort	Zusatzinformation zur Adressierung einer Applikation innerhalb eines Rechners in der TCP/IP Kommunikation. Die Port-Nummer ist frei wählbar in dem Bereich von 1000 bis 64000. Es wird empfohlen die Port-Nummer 3011 für die Maschinensteuerungen zu verwenden. In der RPC SINUMERIK-Konfiguration wird diese Information mit Machine-EndPoint bezeichnet.	3011
MachineTimeout	Mit diesem Wert wird das zeitliche Verhalten beim Versenden von RPC's an RPC SINUMERIK beeinflusst. Kann ein RPC an RPC SINUMERIK nicht zugestellt werden - weil z.B. die Steuerung nicht eingeschaltet ist - dann wird die Zeit bis der RPC-Aufruf mit einem Fehler abgebrochen wird, durch den TimeOut-Wert definiert. Das Attribut darf einen Wert zwischen 0 und 9 annehmen. Diese Werte sind keine Zeitwerte, sonder relative Werte die im Microsoft RPC-System definiert sind. 0 - Min TimeOut 5 - Default TimeOut 9 - Max. TimeOut Es wird empfohlen den Default TimeOut Wert (5) zu verwenden.	5
HostID	Name des Leitrechner in der RPC SINUMERIK-Konfiguration. Frei wählbare Bezeichnung des Leitrechners (Host). Diese Information wird in jedem RPC übergeben zur Identifizierung der Kommunikationspartner.	H1
HostPort	Zusatzinformation zur Adressierung einer Applikation innerhalb eines Rechners in der TCP/IP Kommunikation. Die Port-Nummer ist frei wählbar in dem Bereich von 1000 bis 64000. Es wird empfohlen die Port-Nummer 3010 für den Leitrechner zu verwenden. In der RPC SINUMERIK-Konfiguration wird diese Information mit Host-EndPoint bezeichnet.	3010

Attribute	Bedeutung	Beispiel
HostEnabled	Dieses Attribut zeigt die Bereitschaft der Komponente RPC's zu empfangen an. True - RPC SINUMERIK-OCX ist bereit RPC's zu empfangen. False - RPC SINUMERIK-OCX ist nicht bereit RPC's zu empfangen. Die Bereitschaft RPC's zu empfangen kann durch das Setzten dieses Attributes auf True in der Anwender-Applikation erreicht werden. Die Bereitschaft RPC's zu empfangen wird ebenfalls gesetzt, wenn mindesten ein RPC erfolgreich an RPC SINUMERIK übertragen wurde.	True

Die Attribute HostID, HostPort und HostEnabled gelten für alle Instanzen der RPC SINUMERIK-OCX innerhalb einer Applikation (EXE). Eine Änderung in einer Instanz hat eine Änderung in allen Instanzen zu Folge.

10.3.3 Methoden zum Versenden von RPC's an RPC SINUMERIK

Die RPC SINUMERIK-OCX kapselt die RPC-Schnittstelle von RPC SINUMERIK in COM-Aufrufe. Ein RPC an RPC SINUMERIK wird ausgelöst durch einen Aufruf einer gleichnamigen Methode an der RPC SINUMERIK-OCX Komponente.

Zum Beispiel das RPC

Ret = T_MACHINE_M (Host, Machine, OrderNum)

wird durch die Methode T_MACHINE_M an einer der Instanzen der OCX abgebildet.

Ret = Machine1.T_MACHINE_M (OrderNum)

Bei allen diesen Methoden entfallen die ersten zwei Parameter (Host und Machine). Diese Parameter werden aus den Attributen: HostID und MachineID der jeweiligen Instanz übernommen.

Die Rückgabewerte, die von Methoden geliefert werden, werden im Kapitel: „10.3.6 Fehlerbehandlung“ beschrieben.

10.3.4 Aktivieren der Empfangsbereitschaft

Die RPC SINUMERIK-OCX ist bereit RPC's vom RPC SINUMERIK zu empfangen, wenn mindestens eine RPC erfolgreich an die Maschine gesendet wurde oder das Attribut: **HostEnabled** auf **True** gesetzt wird.

10.3.5 Empfangen der RPC’s vom RPC SINUMERIK

Die RPC’s vom RPC SINUMERIK werden als Ereignisse der jeweiligen Instanz der RPC SINUMERIK-OCX Komponente an die Applikation übergeben.

Zum Beispiel das RPC von RPC SINUMERIK

T_DATA_H (Host, Machine, OrderNum, SFkt, Name1, Name2)

wird an die Applikation als Ereignis

TxDATAxH (OrderNum, SFkt, Name1, Name2)

übergeben.

Da einige Entwicklungssysteme (z.B.: Visual Basic kein „**_**“-Zeichen im Namen der Ereignisse zulassen, wurden die „**_**“-Zeichen in den Namen des RPC’s durch „**x**“ ersetzt.

10.3.6 Fehlerbehandlung

Die Fehlernummern, die von den Methoden der RPC SINUMERIK-OCX geliefert werden können zwei Kategorien unterteilt werden:

- Fehlermeldungen vom Microsoft RPC-System
- Fehlermeldungen vom RPC SINUMERIK

Fehlermeldungen vom Microsoft RPC-System

Diese Fehler treten auf, wenn das Übermitteln eines RPC’s an RPC SINUMERIK oder das aktivieren des RPC-Servers durch die RPC SINUMERIK-OCX nicht durchgeführt werden kann. Die Fehler dieser Kategorie liegen in dem Bereich von 1700 bis 1938 (RPC Status Codes).

Die Beschreibung der in der Praxis auftretenden Fehler aus dieser Gruppe finden Sie unten in der Tabelle "Typische Fehlersituationen". Die vollständige Beschreibung kann der Microsoft Dokumentation entnommen werden.

http://msdn.microsoft.com/library/

unter

Platform SDK -> Networking and Directory services -> Remote Procedure Calls (RPC)

Fehlermeldungen vom RPC SINUMERIK

Diese Fehler treten auf, wenn das Übermitteln eines RPC's an RPC SINUMERIK ordnungsgemäß durchgeführt werden konnte, jedoch - bedingt durch den Inhalt des RPC's bzw. den aktuellen Zustand vom RPC SINUMERIK - eine Korrekte Verarbeitung nicht möglich war. Die Fehler dieser Kategorie liegen im Bereich der negativen Zahlen. ". Die vollständige Beschreibung kann der RPC SINUMERIK Dokumentation „Anhang A - Fehler Nummer" entnommen werden.
Hierbei können folgende Fehlerkonstellationen auftreten:

Typische Fehlersituationen

An dieser Stelle werden einige häufige Fehlersituationen beschreiben:

Tabelle 10-3 Typische Fehlersituationen

Fehlersituation	Rückgabewert einer RPC-Methode
Falsche Wert des Attributes: MachineIP	1722 (RPC_S_SERVER_UNAVAILABLE)
Falsche Wert des Attributes: MachinePort	1722 (RPC_S_SERVER_UNAVAILABLE)
Falsche Wert des Attributes: MachineID	-100 (ERR_WRONG_MACHINE)
Falsche Wert des Attributes: HostID	-110 (ERR_WRONG_HOST)
Falsche Wert des Attributes: HostPort	0, jedoch keine Antwort vom RPC SINUMERIK Gleichzeitig wird auf der Steuerung ein Eintrag in der MCIS_RPCERR.LOG erstellt
Die Empfangsbereitschaft der RPC SINUMERIK-OCX Komponente kann nicht aktiviert werden.	1720 (RPC_S_CANT_CREATE_ENDPOINT) Diese Fehler tritt meistens auf, wenn bereits eine andere Applikation auf dem gleichen Rechner bereits die im dem Attribut HostPort vorgegebenen Port nutzt. Z. B.: Es wird versucht eine Applikation zu starten, die RPC SINUMERIK-OCX verwendet und gleichzeitig wird die Applikation RPC SINUMERIK Test - mit dem gleichen HostPort Wert - ausgeführt
Unterbrechung der TCP/IP Verbindung	1722 oder 1726 (nach ca. 20 Sek.) Nach Wiederherstellung der Verbindung liefern die RPC-Aufrufe wieder 0

10.3.7 Einschränkungen im Zusammenhang mit Test

Innerhalb der Entwicklungsumgebung von VisualBasic werden Ereignisse von der OCX nicht ausgelöst, wenn VisualBasic durch ein Haltepunkt unterbrochen ist.

10.4 Testapplikation RPC SINUMERIK Test

Die Applikation RPC SINUMERIK Test ermöglicht es im Dialog die einzelnen RPC's der RPC SINUMERIK-Schnittstelle zu senden bzw. zu Empfangen. Es kann eine Verbindung zu Mehreren Steuerungen aufgebaut werden.

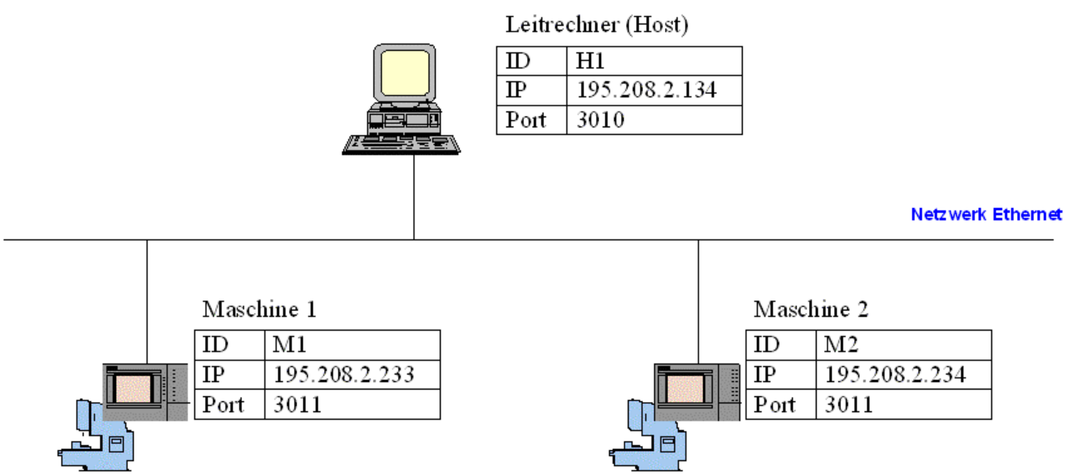
Die Applikation wurde mit Visual Basic 6.0 erstellt. Der Source-Code der Applikation wird in dem Verzeichnis Siemens\MCIS\RPC SINUMERIK\ RPC SINUMERIK Test ausgeliefert. (siehe auch Kap.: 10.4.4 Quell-Code der Applikation RPC SINUMERIK Test)

Um RPC SINUMERIK Test zu verwenden müssen Voraussetzungen erfühlt werden:

1. Es muss mindestens eine SINUMERIK 840D Steuerung mit installiertem RPC SINUMERIK Paket zur Verfügung stehen.
2. Auf dem Leitrechner (Windows-PC) muss das TCP/IP-Protokoll installierst werden.
3. Es muss eine Netzwerk-Verbindung zwischen den Steuerungen und dem Leitrechner vorhanden sein.

10.4.1 Konfiguration

Die Konfiguration wird am Beispiel folgender Netzwerk Architektur dargestellt

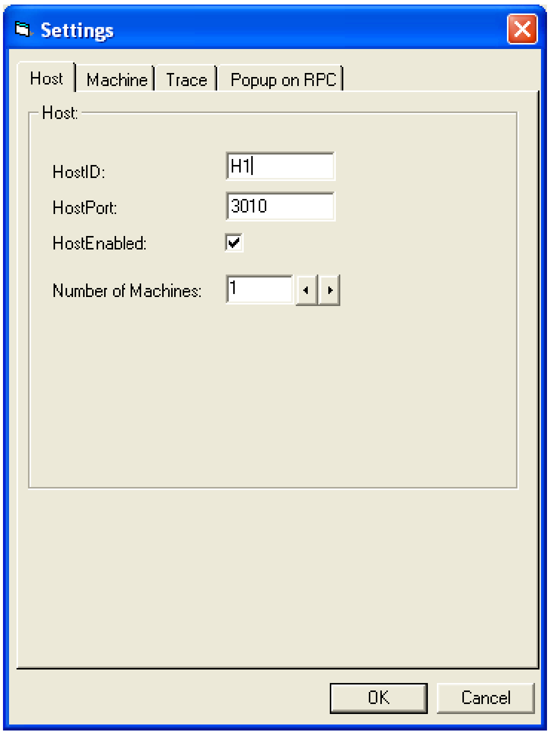


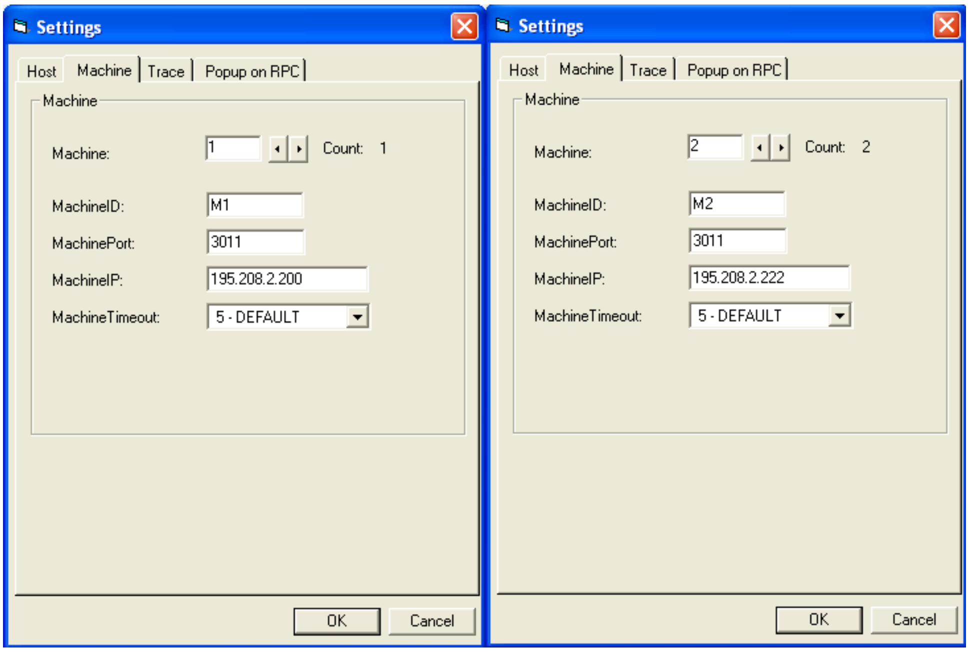
Für jeden der beteiligten Rechner werden folgende Information benötigt:

Tabelle 10-4 Benötigte Informationen bei Konfiguration der Rechner

ID	Frei wählbare Bezeichnung der Kommunikationspartner. Diese Information wird in jedem RPC übergeben zur Identifizierung der Kommunikationspartner.
IP	IP-Netzwerkadresse des Leitrechners oder Steuerung. Diese Information kann der Windows-Netzwerkinstallation entnommen werden. Es wird eine feste IP-Vergabe vorausgesetzt. In der RPC SINUMERIK-OCX kann an dieser Stelle alternativ der Netzwerkname des jeweiligen Rechners verwendet werden.
Port	Zusatzinformation zur Adressierung einer Applikation innerhalb eines Rechners in der TCP/IP Kommunikation. Die Port-Nummer ist frei wählbar in dem Bereich von 1000 bis 64000. Es wird empfohlen die Port-Nummer 3010 für Leitrechner und 3011 für die Steuerungen zu verwenden. In der RPC SINUMERIK-Konfiguration wird diese Information mit EndPoint bezeichnet.

Aus der dargestellten Netzwerk Architektur resultieren folgende Einträge in den Konfigurations-Masken von RPC SINUMERIK Test. Die Konfigurations-Masken sind über das Menü **Settings** aufrufbar.

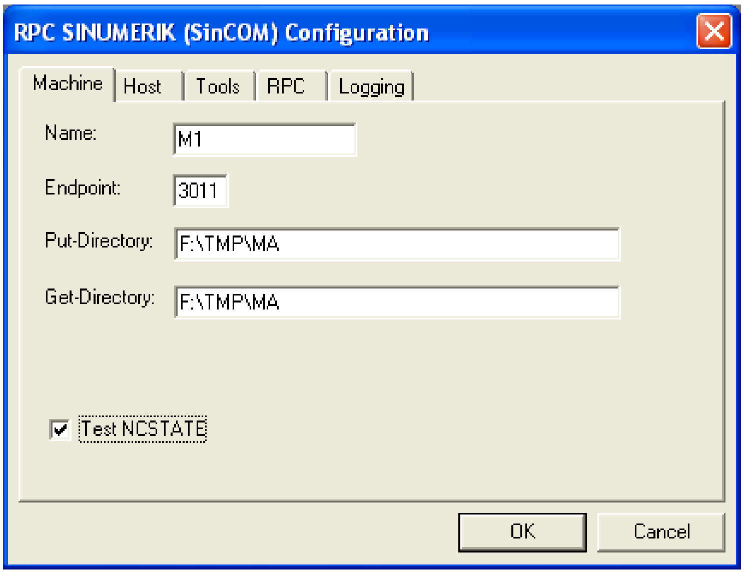


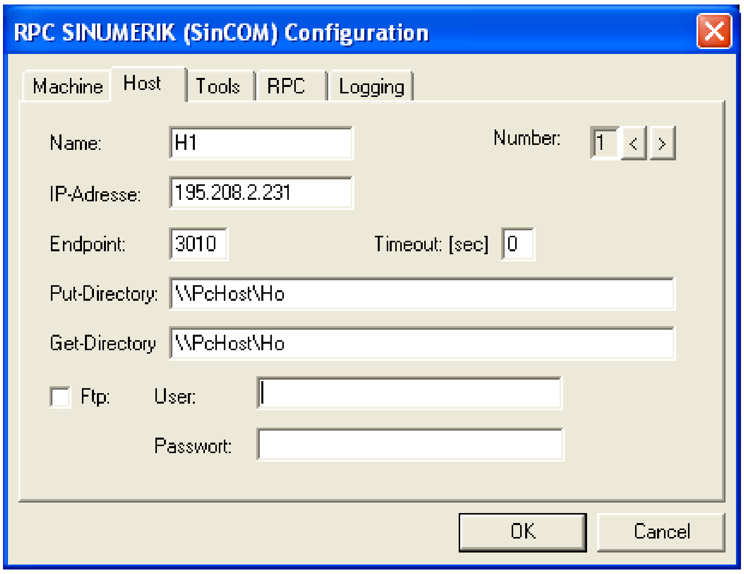


Die Bedeutung der Eingabefelder HostEnabled und MachineTimeout entspricht den gleichnamigen Attributen der RPC SINUMERIK-OCX.

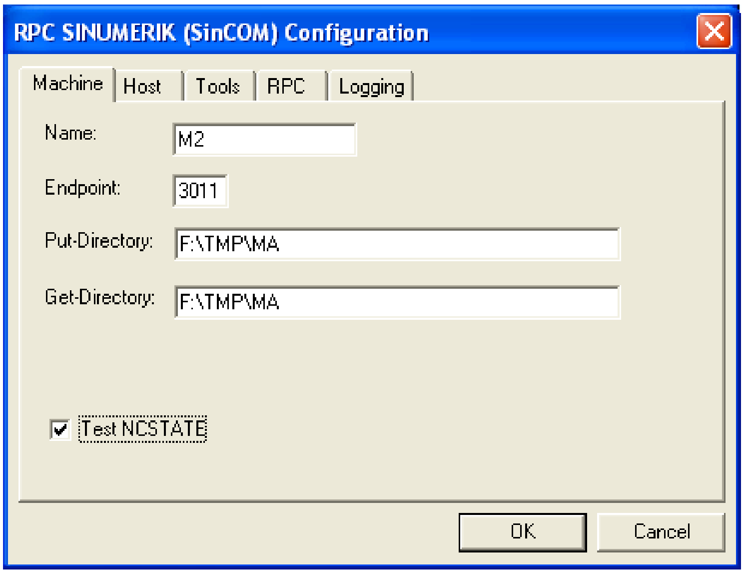
Die Konfiguration der Verbindung an den Steuerungen erfolgt mit der Applikation **scconfig**. Entsprechen der dargestellten Netzwerk Architektur müssen folgende Einträge vorgenommen werden. Eine weitergehende Beschreibung zur scconfig entnehmen Sie bitte den vorderen Kapiteln.

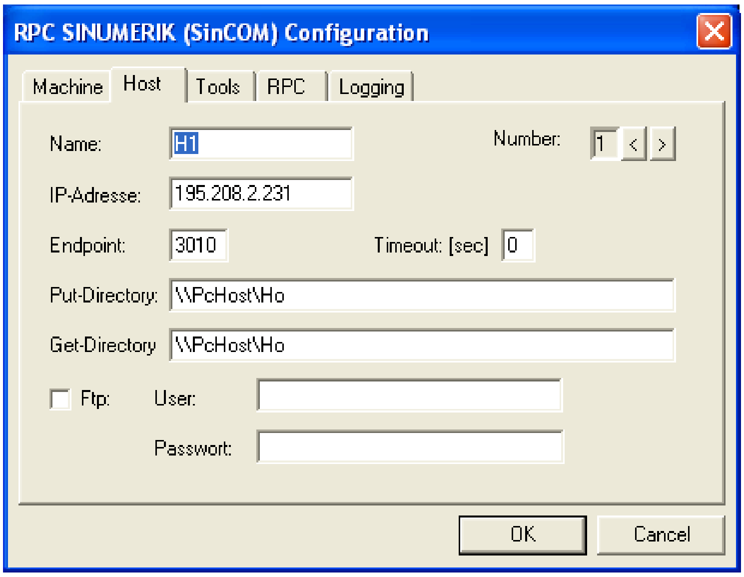
Maschine 1



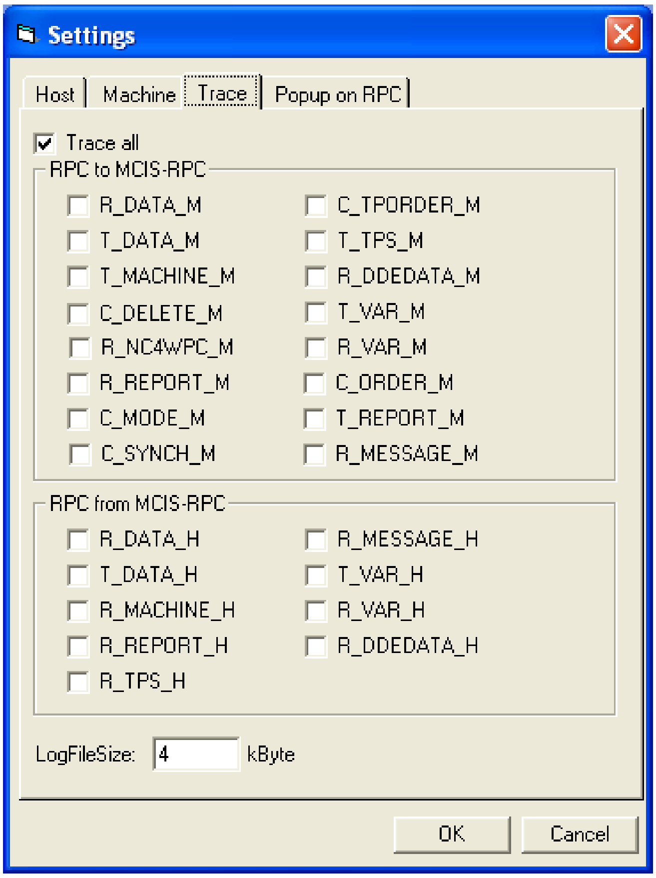


Maschine 2

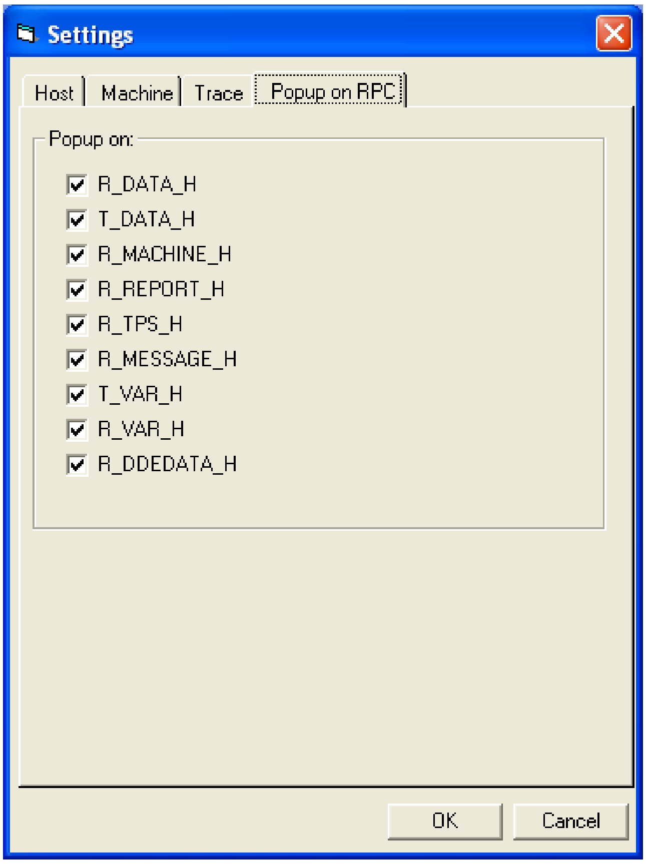

RPC SINUMERIK (SinCOM) Configuration
Machine
Host
Tools
RPC
Logging
Name:
M2
Endpoint:
3011
Put-Directory:
F:\TMP\MA
Get-Directory:
F:\TMP\MA
Test NCSTATE
OK
Cancel


RPC SINUMERIK (SinCOM) Configuration
Machine
Host
Tools
RPC
Logging
Name:
H1
Number:
1
IP-Adresse:
195.208.2.231
Endpoint:
3010
Timeout: [sec]
0
Put-Directory:
\\PcHost\Ho
Get-Directory
\\PcHost\Ho
Ftp:
User:
Passwort:
OK
Cancel

Über die Register-Karte „**Trace**“ kann gezielt selektiert werden, welche RPC’s protokolliert werden. Die Protokollausgaben erfolgen auf dem Bildschirm und in die Logdatei RpcSinumerikTest.LOG, deren maximal Größe hier vorgegeben werden kann.

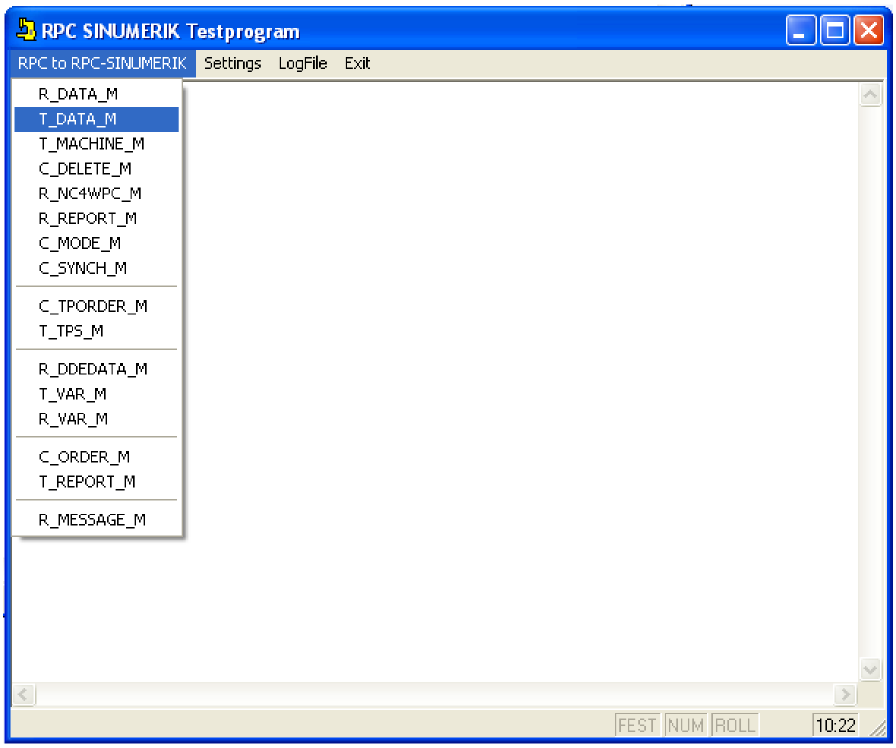


Über die Register-Karte „**Popup on RPC**“ kann die Reaktion auf Empfangene RPC’s definiert werden. Standardmäßig wird als Reaktion auf ein Empfangenes RPC eine Dialogmaske mit sämtlichen Daten des RPC aufgeblendet.

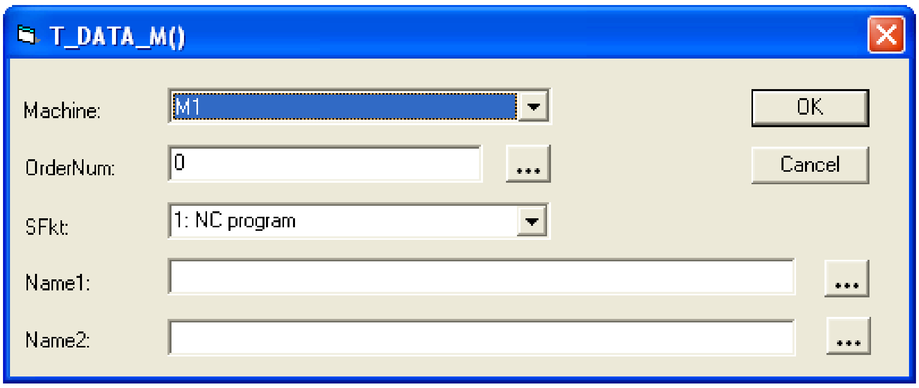


10.4.2 Senden von RPC's an RPC SINUMERIK

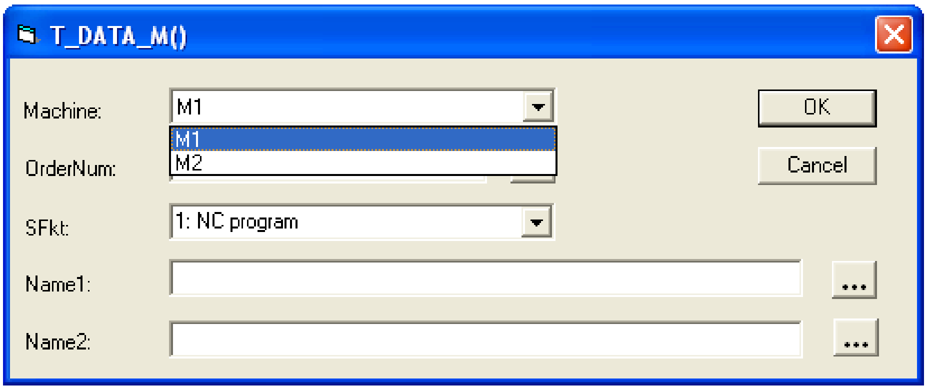
Die Applikation RPC SINUMERIK Test ermöglicht es alle definierten RPC's über eine Dialogmaske an RPC SINUMERIK zusenden. Die Dialogmasken werden über das Menü: „**RPC to RPC SINUMERIK**" aufgerufen.



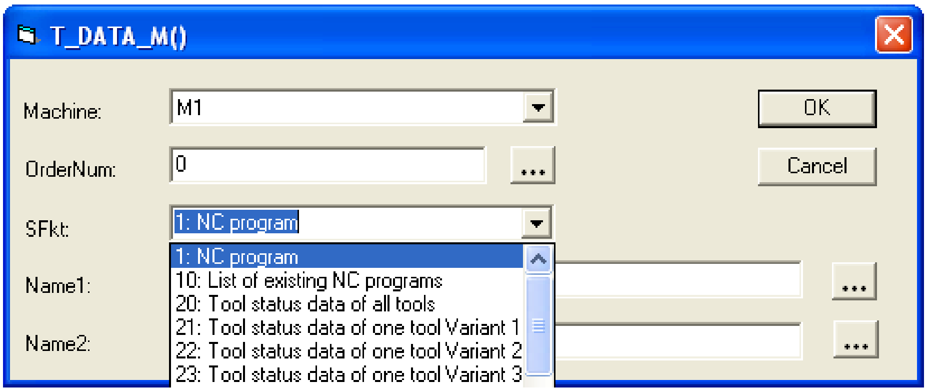
Hier als Beispiel die Maske zum Absenden des RPC: **T_DATA_M**.



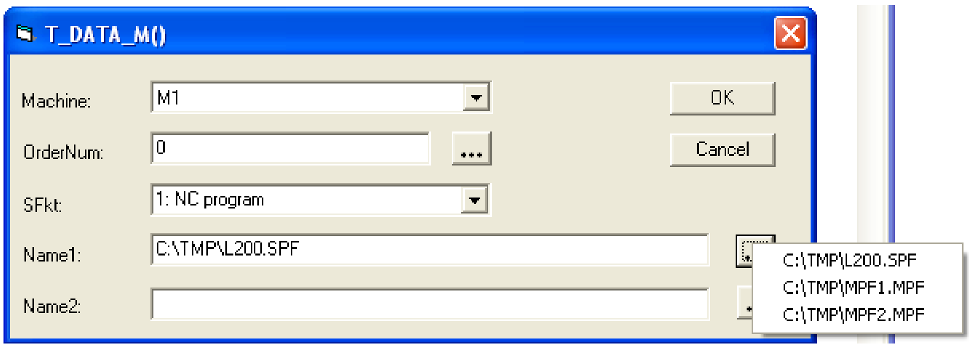
Die Eingabefelder sind mit den gleichen Namen bezeichnet wie die entsprechenden RPC's Parameter in der RPC SINUMERIK-Dokumentation benannt sind. Die Bedeutung dieser Parameter ist dieser Dokumentation zu entnehmen. Mit dem Parameter *Machine* kann vorgegeben werden, an welche Steuerung das RPC gesendet wird. Dieses Eingabefeld ist als Auswahlfeld realisiert. Es schlägt die in der Konfiguration definierten Kommunikationspartnern vor.



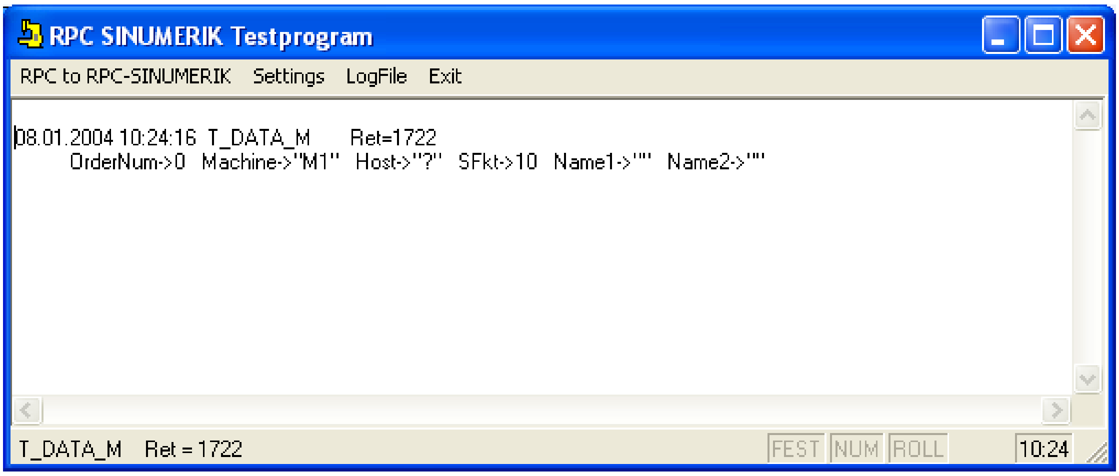
Alle Parameter, die in der RPC SINUMERIK-Dokumentation eine vordefinierte Werte annehmen können (z.B. SFkt) sind als Auswahlfelder realisiert.



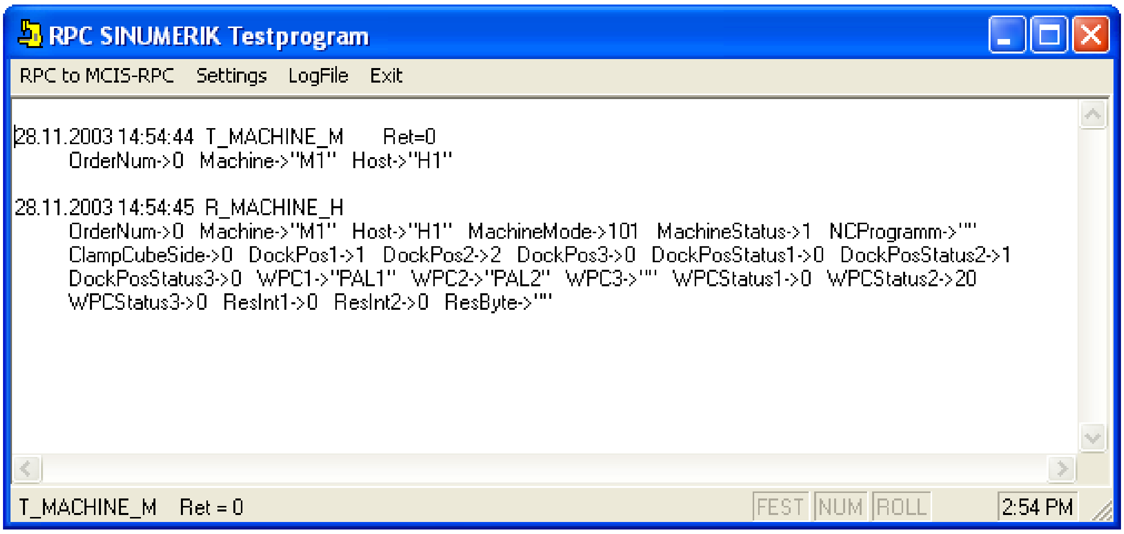
Zu allen anderen Eingabefeldern kann über die Schaltfläche „...“ ein bereits früher verwendeter Wert ausgewählt werden.



Mit der Schaltfläche „OK“ wird das RPC gesendet. Alle gesendeten RPC’s werden protokolliert. Die Protokollausgabe erfolgen auf dem Bildschirm in dem Hauptfenster der Applikation RPC SINUMERIK Test und in der Logdatei: RpcSinumerikTest.LOG. Zu jedem gesendetem RPC wird ebenfalls das Rückgabewerte mit protokolliert und in der Statuszeile dargestellt. (Ret=0). Die Bedeutung der Rückgabewerte wird im Kapitel: „10.3.6 Fehlerbehandlung" beschrieben.



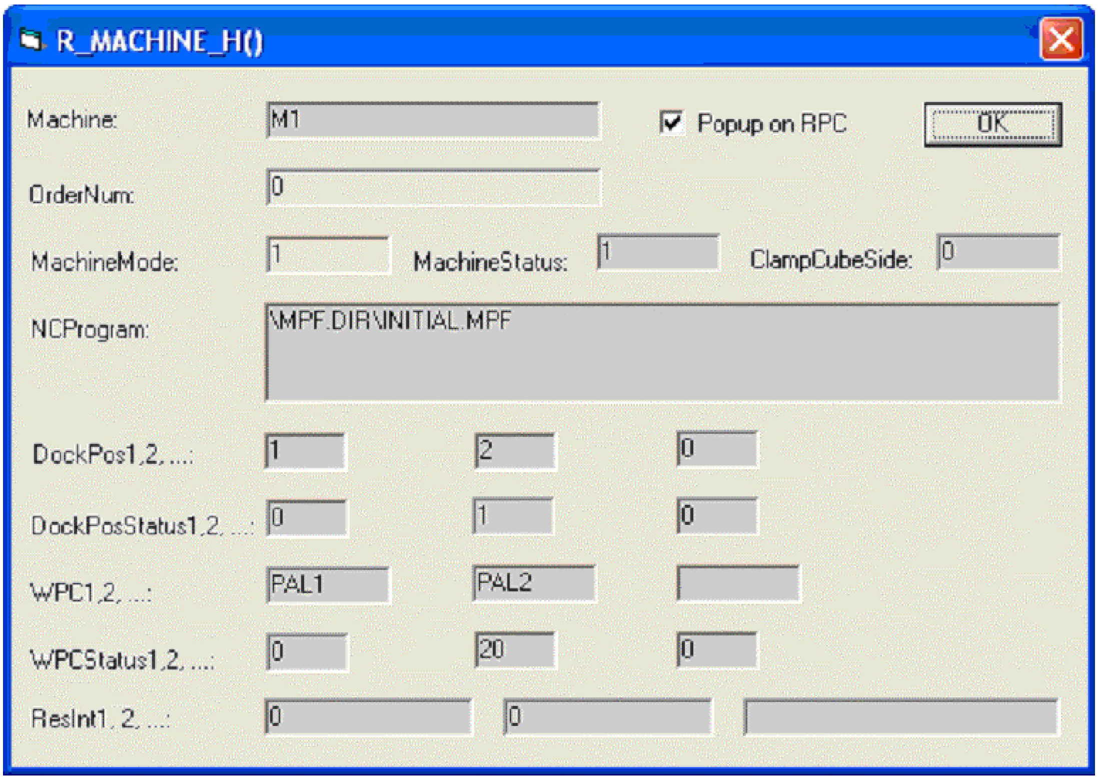
Um die grundsätzliche Verbindung zu RPC SINUMERIK zu Testen kann das RPC: T_MACHINE_M gesendet werden. RPC SINUMERIK antwortet daraufhin mit dem RPC: R_MACHINE_H.



10.4.3 Empfangen von RPC’s vom RPC SINUMERIK

Nach der Konfiguration der Kommunikation Partner - mit der aktivierten Option **HostEnabled** ist die Applikation RPC SINUMERIK Test bereit RPC’s von RPC SINUMERIK zu empfangen. Wird die Option HostEnabled nicht aktiviert in der Konfiguration, dann werden RPC’s vom RPC SINUMERIK erst dann empfangen, wenn zumindest ein RPC erfolgreich an RPC SINUMERIK übertragen wurde.

Die empfangenen RPC’s werden protokolliert. Die Protokollausgabe erfolgen auf dem Bildschirm in dem Hauptfenster der Applikation RPC SINUMERIK Test und in der Logdatei: RpcSinumerikTest.LOG. Zusätzlich wird zu jedem RPC eine Dialogmaske mit allen Parametern des RPC’s aufgeblendet.



Die Ausgabefelder sind mit den gleichen Namen bezeichnet wie die entsprechenden RPC’s Parameter in der RPC SINUMERIK-Dokumentation benannt sind. Die Bedeutung dieser Parameter ist dieser Dokumentation zu entnehmen.

Das Aufblenden der Dialogmasken zu den empfangenen RPC’s kann über das Optionsfeld: **„Popup on RPC“** in der jeweiligen Maske oder in der Konfiguration (Registerseite: Popup on RPC) ein- bzw. ausgeschaltet werden.

10.4.4 Quell-Code der Applikation RPC SINUMERIK Test

Der Quell-Code der Applikation wird vom Setup im Verzeichnis Siemens\MCIS\RPC SINUMERIK\RPC SINUMERIK Test hinterlegt.
Die Applikation wurde mit dem Entwicklungssystem Microsoft VisualBasic 6.0 erstellt.

Die Applikation besteht aus folgenden Modulen:

Tabelle 10-5 Module der Applikation RPC SINUMERIK Test

RpcSinumerikTest.vbp	VisualBasic Projekt-Datei
RpcSinumerikTest.frm **RpcSinumerikTest.frx**	Hauptfenster der Applikation
RpcSinumerikConfig.frm **RpcSinumerikConfig.frx**	Konfiguration Maske
History.frm	Auswahl der Eingabewerte
Logen.bas	Protokollier-Funktionen
Util.bas	Hilfsfunktionen
R_DATA_H.frm **...**	Eingabe- bzw. Anzeigemasken zu den entsprechenden RPC's

10.5 Beispiele der Anwendung der RPC SINUMERIK-OCX

In allen Beispielen die in diesem Kapitel vorgestellt werden, wird die Netzwerkkonfiguration verwendet, die schon im Kapitel: „10.4.1 Konfiguration" dargestellt wurde. Um die Beispiele an Ihre Netzwerkkonfiguration anzupassen müssen die IP-Adressen im Quell-Code der Beispiele entsprechen verändert werden. Es wird auch vorausgesetzt, dass RPC SINUMERIK auf den Maschinen-steuerungen installiert ist und eine Netzverbindung hergestellt werden kann.

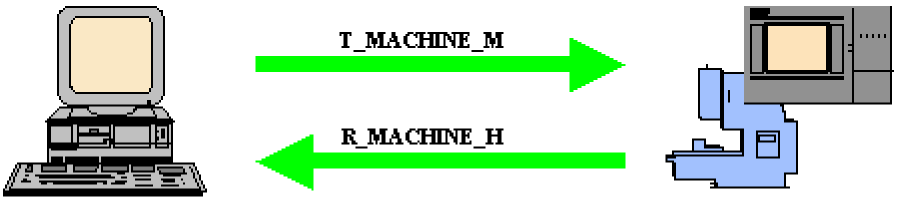
Hinweis

Die Stringbegrenzung '\0' in den Beispielen der Nahtstellenbeschreibung zum Fertigungsleitrechner (Kapitel 5), ist nur bei Applikationen die mit C++ erstellt werden, notwendig.

10.5.1 Beispiel 1 - Abfrage des Maschinenzustandes (Visual Basic)

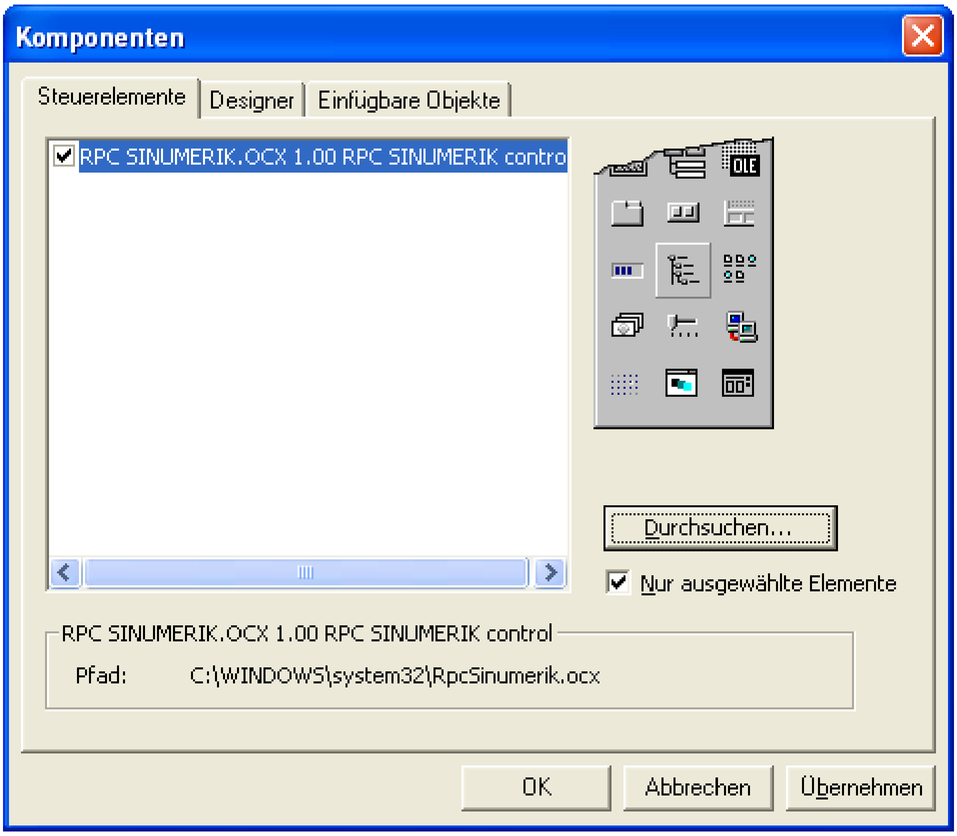
Hier ein Beispiel einer Visual Basic Anwendung, die das RPC: T_MACHINE_M() an die Maschine sendet und die Reaktion darauf in Form vom RPC: R_MACHINE_H() vom RPC SINUMERIK empfängt. Die Verwendung der RPC's ist in der RPC SINUMERIK-Dokumentation in den Kapiteln: 5.3.1 und 5.3.2 beschrieben.

Es werden alle Schritte dargestellt, die innerhalb der Visual Basic Entwicklungsumgebung notwendig sind, um die Applikation zu erstellen.

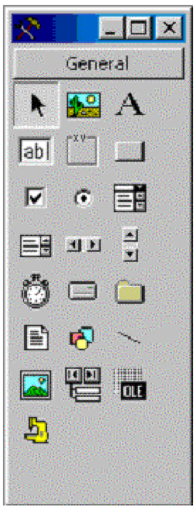


Einbindung der RPC SINUMERIK-OCX Komponente in Visual Basic 6.0

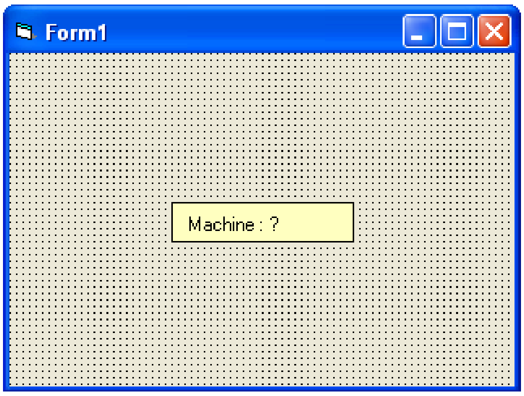
Bevor die RPC SINUMERIK-OCX Komponente innerhalb vom Visual Basic verwendet werden kann, muss sie dort bekannt gemacht werden. Dies erfolgt über das Menü: Projekt -> Komponenten.



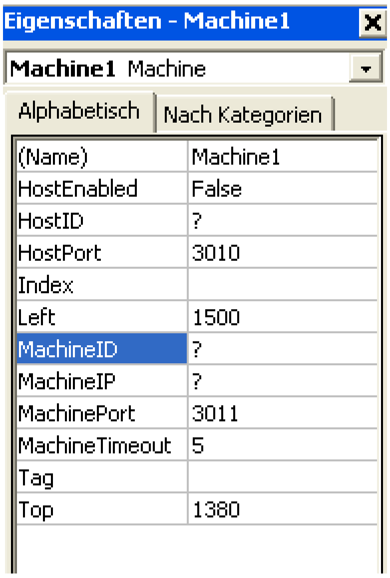
Ab diesem Zeitpunkt erscheint die Ikone der RPC SINUMERIK-OCX im Toolbox-Fenster (gelbe Maschine)



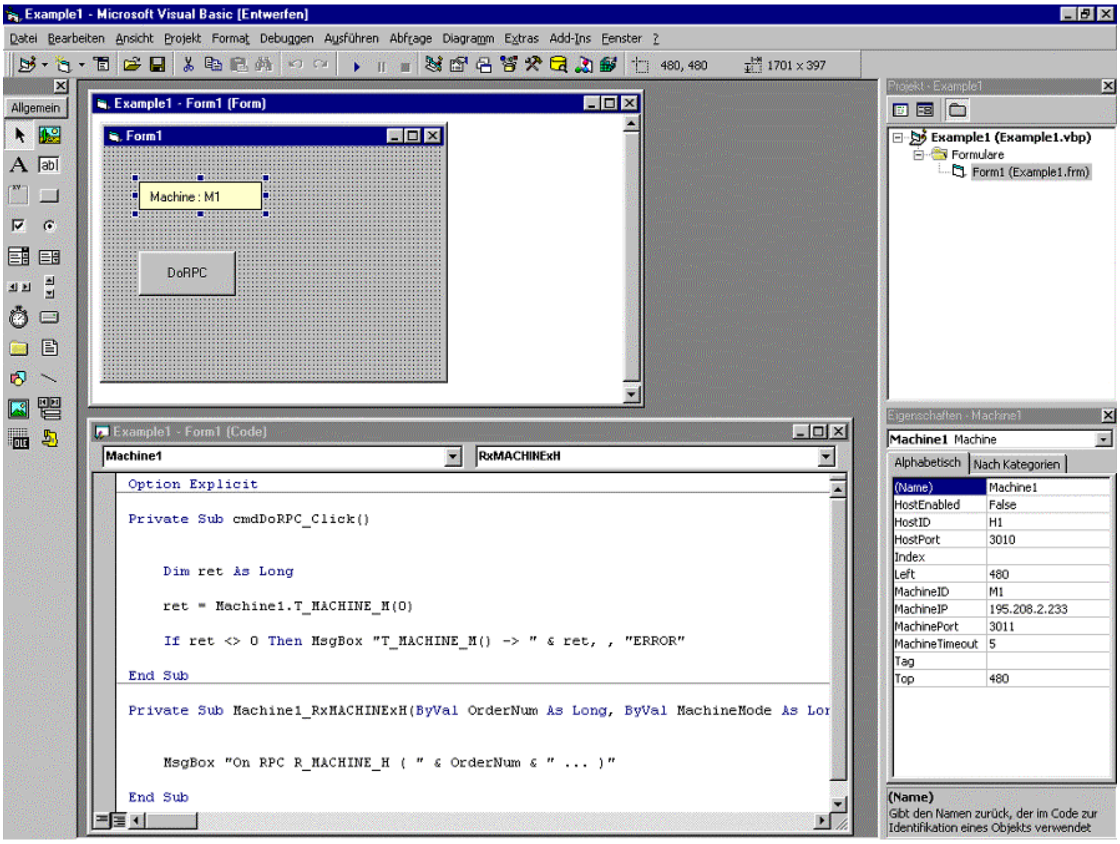
Mit Drag&Drop kann jetzt die RPC SINUMERIK-OCX in einer Form plaziert werden. Die Komponente ist innerhalb der Entwicklungsumgebung sichtbar. Jedoch ist sie zur Laufzeit einer fertigen Applikation nicht mehr sichtbar.



In dem Properties-Fenster können jetzt Attribute hinterlegt werden. Die IP-Adresse der Maschinensteuerung muss entsprechend Ihrer Netzwerkkonfiguration eingesetzt werden.

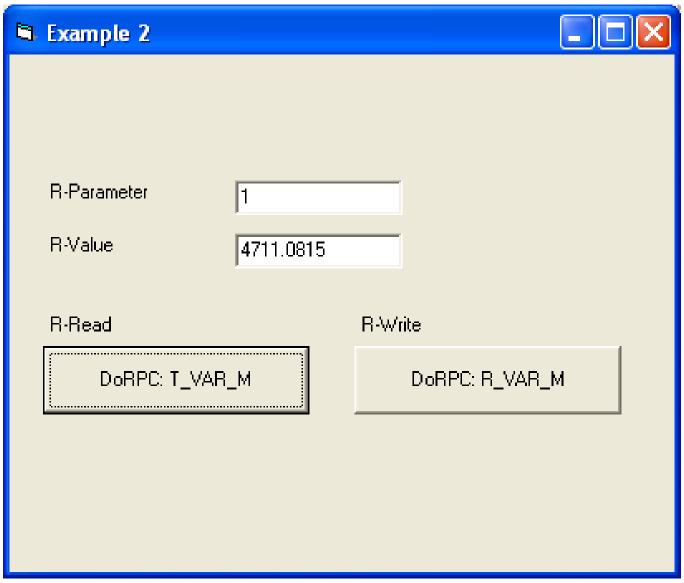


Mit der Änderung des Attributes: **MachineID** ändert sich auch die Darstellung der OCX in der Form. Das Absenden des RPC's **T_MACHINE_M** wird durch das Betätigen der Schaltflächen „DoRPC“ ausgelöst. Die RPC SINUMERIK-Applikation antwortet mit dem RPC **R_MACHINE_H**. Als Reaktion darauf wird in der Beispielapplikation eine Message-Box aufgeblendet.

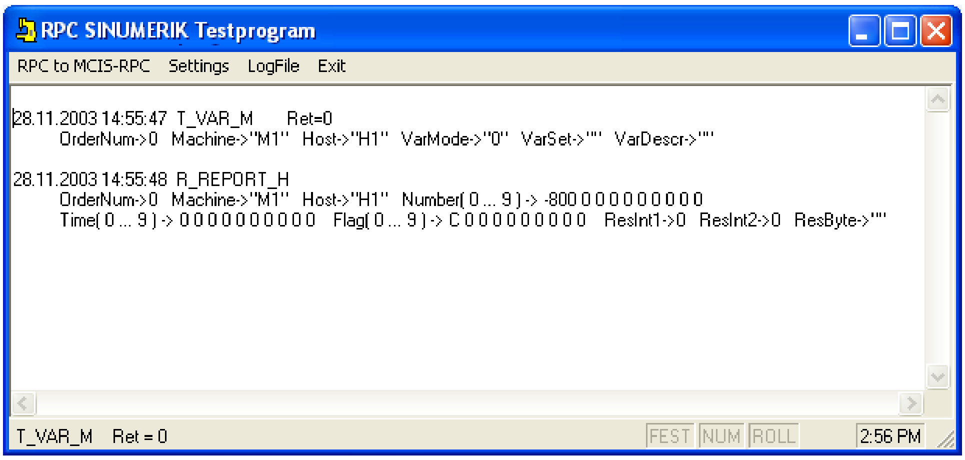

Example1 - Microsoft Visual Basic [Entwerfen]
Datei Bearbeiten Ansicht Projekt Format Debuggen Ausführen Abfrage Diagramm Extras Add-Ins Fenster ?
480, 480
1701 x 397
Allgemein
Example1 - Form1 (Form)
Form1
Machine : M1
DoRPC
Projekt - Example1
Example1 (Example1.vbp)
Formulare
Form1 (Example1.frm)
Example1 - Form1 (Code)
Machine1
RxMACHINExH
Option Explicit
Private Sub cmdDoRPC_Click()
Dim ret As Long
ret = Machine1.T_MACHINE_M(0)
If ret <> 0 Then MsgBox "T_MACHINE_M() -> " & ret, , "ERROR"
End Sub
Private Sub Machine1_RxMACHINExH(ByVal OrderNum As Long, ByVal MachineMode As Lor
MsgBox "On RPC R_MACHINE_H (" & OrderNum & " ...)"
End Sub
Eigenschaften - Machine1
Machine1 Machine
Alphabetisch
Nach Kategorien
(Name) Machine1
HostEnabled False
HostID H1
HostPort 3010
Index
Left 480
MachineID M1
MachineIP 195.208.2.233
MachinePort 3011
MachineTimeout 5
Tag
Top 480
(Name)
Gibt den Namen zurück, der im Code zur Identifikation eines Objekts verwendet

10.5.2 Beispiel 2 - Lesen und Schreiben von R-Parametern (Visual Basic)

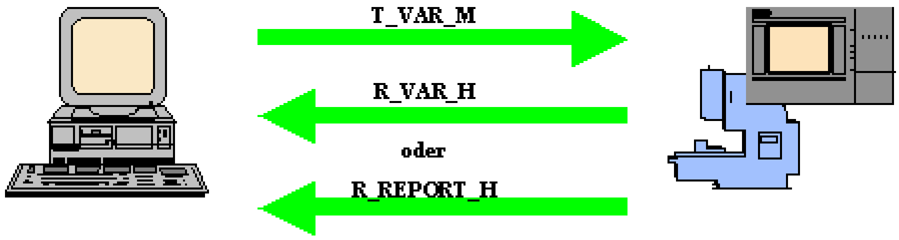
In diesem Beispiel wird die Verwendung des RPC SINUMERIK Variablen-Dienstes am Beispiel von R-Parametern erläutert. Die vollständige Beschreibung des Variablen-Dienstes ist in der RPC SINUMERIK-Dokumentation Kapitel: „7 Projektierbare Datenübertragung/Variablen Dienst“ beschrieben.



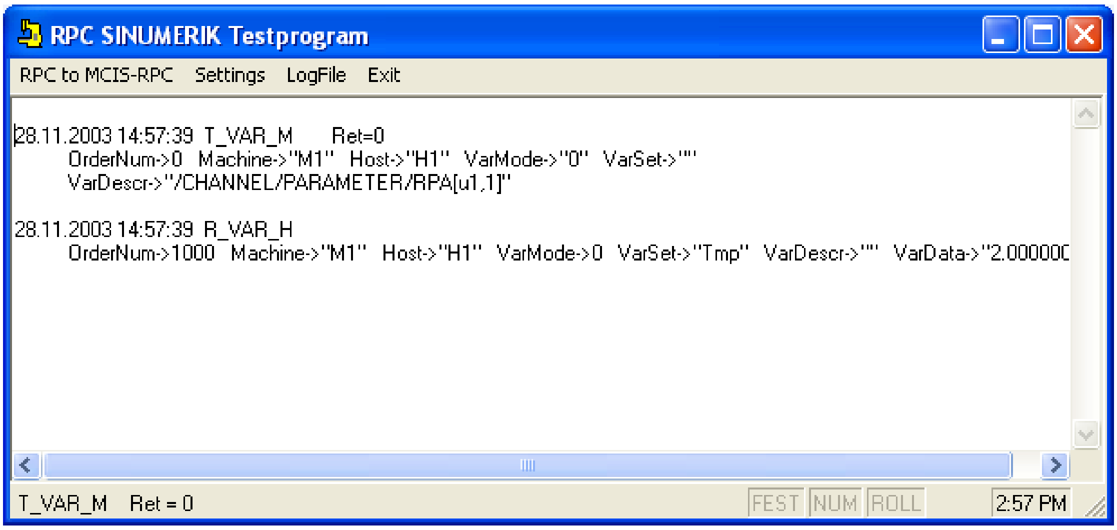
Um den Variablen-Dienst von RPC SINUMERIK zu aktivieren ist es notwendig, dass mindestens ein Variablen-Set in der Datei c:\add_on\scvarset.ini auf der Steuerung definiert ist. Änderungen in dieser Datei sind erst nach einem Neustart der Steuerung wirksam. Is das nicht der Fall, dann antwortet RPC SINUMERIK mit dem RPC: R_REPRT_H() und dem Fehler: -800.



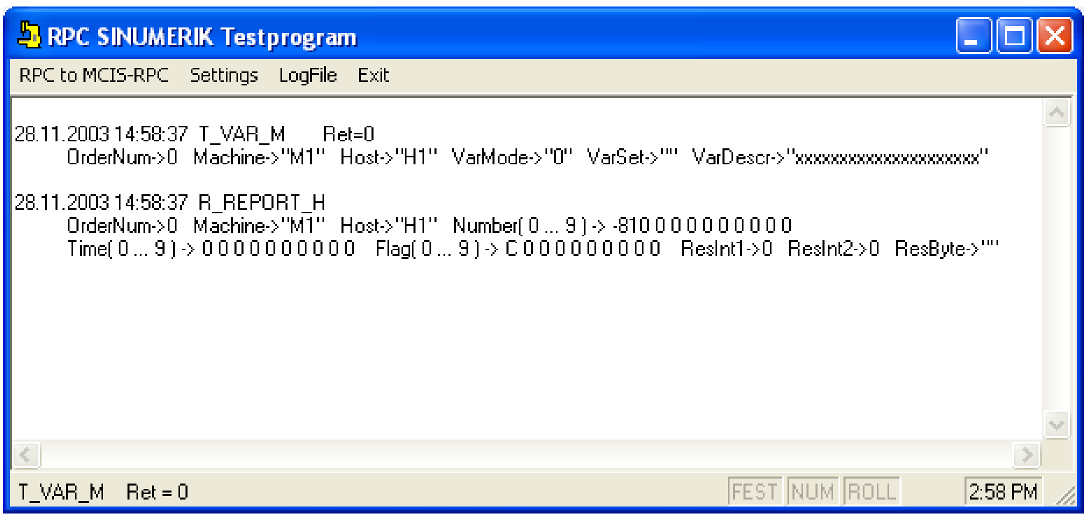
**Ablaufschema für das
Lesen von R-Parametern**



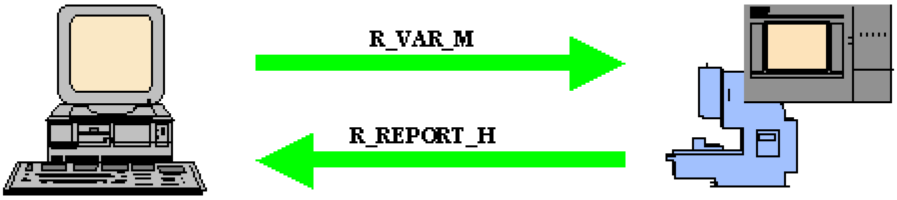
Die VisualBasic Applikation fordert den Wert eines R-Parameters mit dem RPC:
T_VAR_M() an.
RPC SINUMERIK liefert den aktuellen Wert des R-Parameters mit dem RPC:
R_VAR_H().



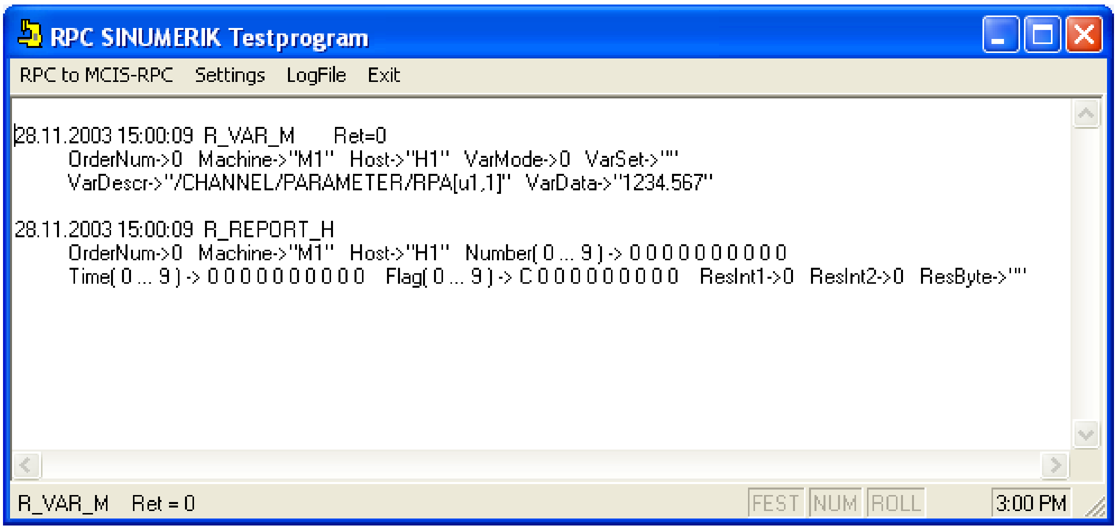
Im Fehlerfall antwortet RPC SINUMERIK mit dem RPC: **R_REPORT_M()**.



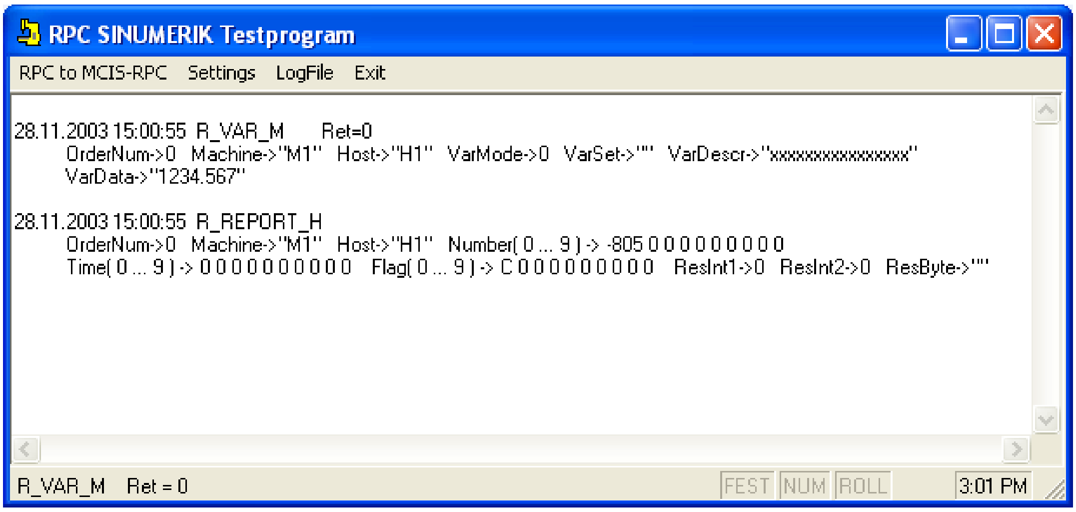
Ablaufschema für das Schreiben von R-Parametern



Die VisualBasic Applikation übergibt die Daten eines R-Parameters mit dem RPC: **R_VAR_M()** an RPC SINUMERIK. RPC SINUMERIK bestätigt den Schreibvorgang mit dem RPC: **R_REPORT_H()**.



Im Fehlerfall wird ebenfalls das RPC: **R_REPORT_H()** von RPC SINUMERIK gesendet. Jedoch enthält der Parameter **Number(0)** entsprechenden Fehler-Code.

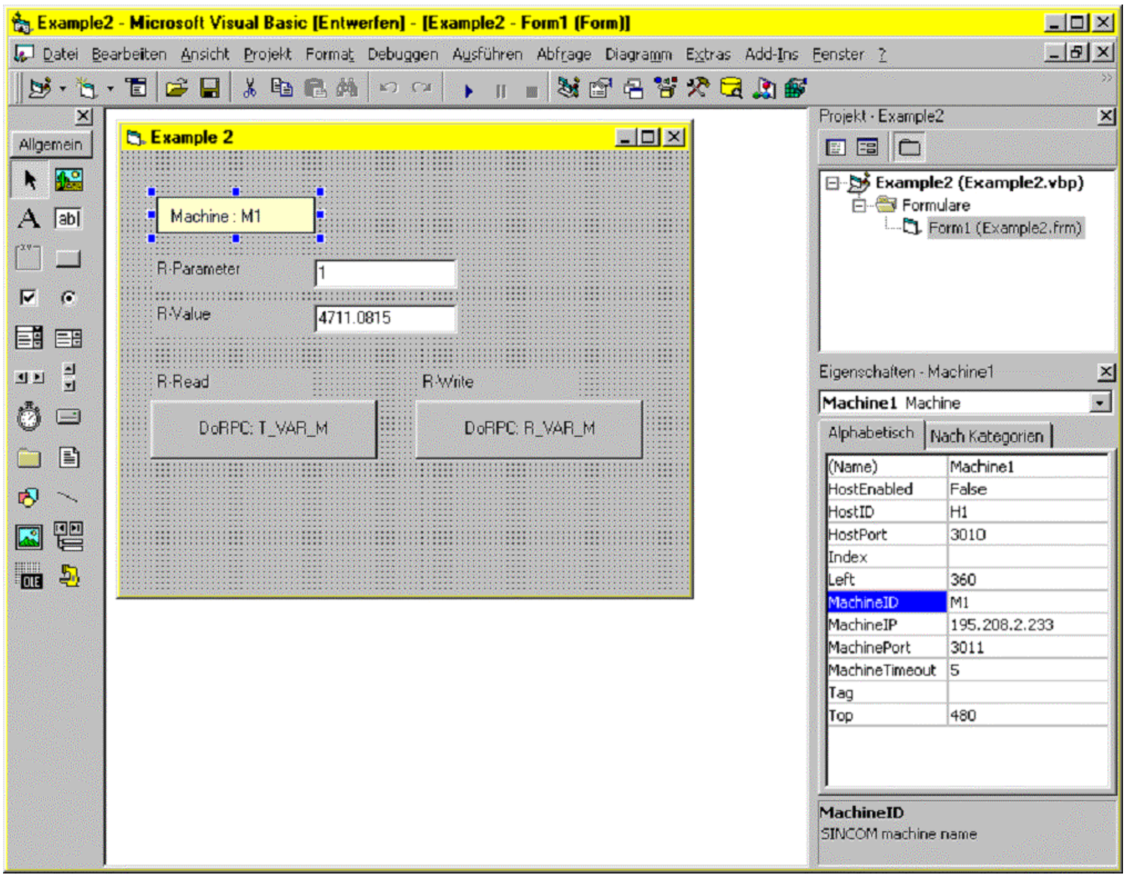


Visual Basic Source-Code

Die Einbindung der RPC SINUMERIK-OCX erfolgt wie im Kapitel: „10.5.1 Beispiel 1 - Abfrage des Maschinenzustandes (Visual Basic) Einbindung der RPC SINUMERIK-OCX Komponente in Visual Basic 6.0“ bereits beschrieben.

Über das Eingabefelde: **R-Parameter** wird die Nummer des R-Parameters für das Lesen und Schreiben vorgegeben. In dem Eingabefeld: **R-Value** wird - beim Lesen - der aktuelle Wert des R-Parameters angezeigt, bzw. - beim Schreiben - der neue Wert eingegeben.

Das Absenden der RPC's T_VAR_M() und R_VAR_M() erfolgt über die entsprechenden Schaltflächen.



Date: \Examples\Example2\Example2.frm

```
Option Explicit

Option Explicit
Private Sub cmdR_VAR_M_Click()

   'write R parameter

   Dim ret            As Long
   Dim RParam         As Long      ' R parameter number
   Dim RItem          As String    ' item for access
   Dim RValue         As String


   RParam = Val(txtRParam.Text)
   RItem = "/Channel/Parameter/R[" & RParam & "]"
   RValue = txtRValue.Text

   ret = Machine1.R_VAR_M(0, 0, "", RItem, RValue)
   If ret <> 0 Then MsgBox "R_VAR_M() -> " & ret, , "ERROR"

End Sub

Private Sub Machine1_RxVARxH(ByVal OrderNum As Long, ByVal VarMode As Long, _
              ByVal VarSet As String, ByVal VarDescr As String, ByVal VarData As String)

   'show R parameter in the form

   txtRValue.Text = VarData

End Sub

Private Sub Machine1_RxREPORTxH(ByVal OrderNum As Long, ByVal Typ As Long, _
              ByVal Number As Variant, ByVal Time As Variant, ByVal Flag As Variant, _
              ByVal ResInt1 As Long, ByVal ResInt2 As Long, ByVal ResByte As String)

   If Number(0) <> 0 Then
      MsgBox "On RPC R_REPEOR_H ( ... Number(0)->" & Number(0) & " )"
   End If

End Sub
```

10.5.3 Beispiel 3 - Aktives-Lesen von R-Parametern (Internet Explorer)

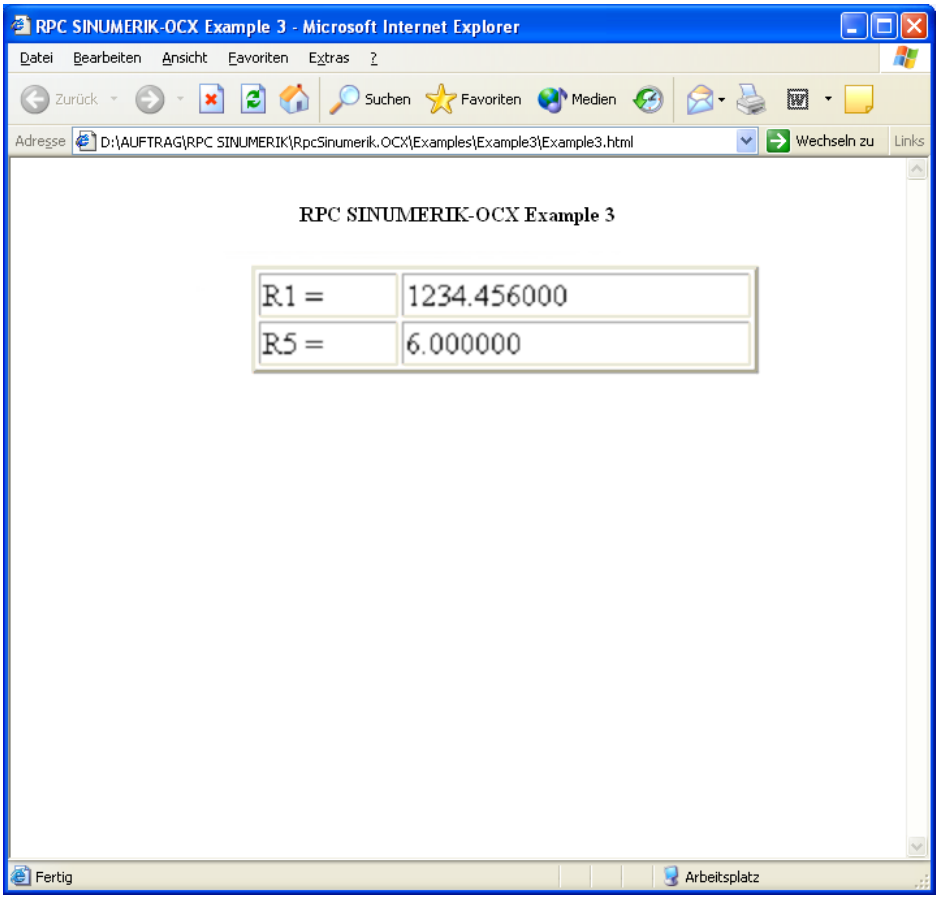
In diesem Beispiel wird das Aktives-Lesen von R-Parametern unter Verwendung vom MS-Internet Explorer demonstriert. Die Funktion Aktives-Lesen (auch als Hotlink bezeichnet) ermöglicht es, dass die RPC SINUMERIK-OCX über jede Änderung der Daten aus einem Variablen-Set vom RPC SINUMERIK unmittelbar informiert wird.

In diesem Beispiel wird ein Variablen-Set mit dem Namen „Set01" bestehend aus den R-Paremeter R1 und R5 verwendet.
Die Definition des Variablen-Set's erfolgt in der Datei : c:\add_on\scvarset.ini auf der Steuerung.

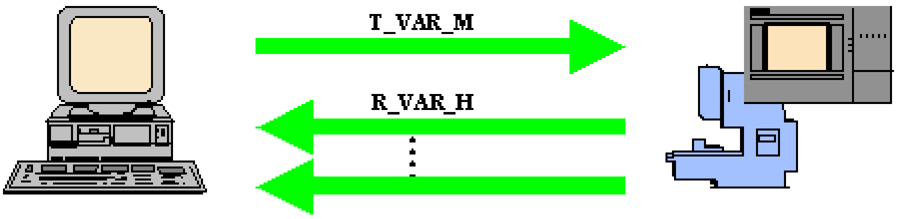
```
[Set01]
Mode=0
Host=FLR1
Var01=/Channel/Parameter/R[1]
Var02=/Channel/Parameter/R[5]
```

Die Verwendung vom MS-Internet Explorer setzt voraus, dass die RPC SINUMERIK-OCX bereits installiert ist.

Datei: \Examples\Example3\Example3.html dargestellt mit MS-Internet Explorer

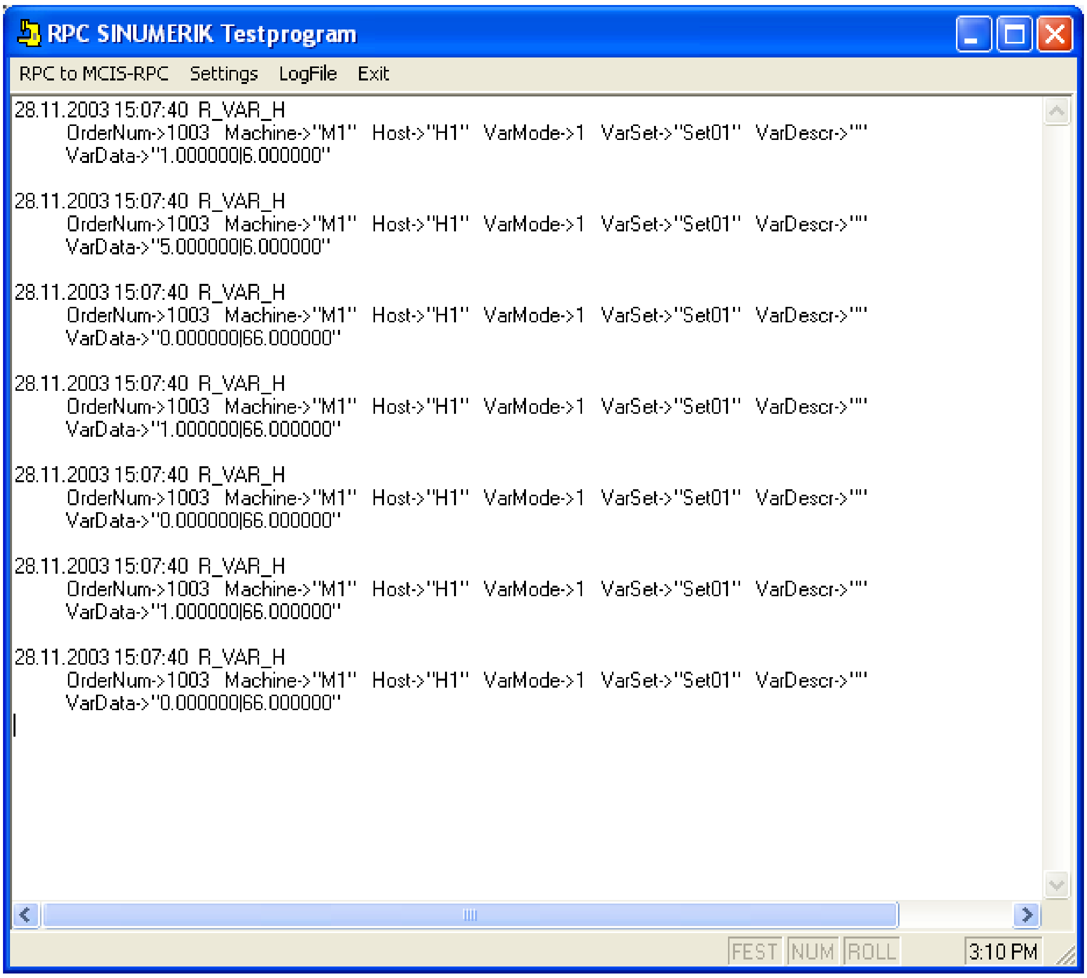


Ablaufschema für das
Aktive-Lesen von R-Parametern



Über eine VBScript-Funktion wird zum Zeitpunkt des Ladens der HTML-Seite das RPC: **T_VAR_M()** an RPC SINUMERIK gesendet. Mit diesem RPC werden die aktuellen Werte der Variablen im Sets: „**Set01**“ angefordert. Diese Daten werden unmittelbar vom RPC SINUMERIK mit dem RPC: **R_VAR_H()** geliefert.

Bei Änderung der Variablen aus dem Set (R1 oder R5) meldet RPC SINUMERIK die Änderung mit dem RPC: **R_VAR_H().**



Source-Code der HTML-Seite

Das RPC SINUMERIK-OCX wird über das <OBJECT>-Tag im HTML-Code eingebunden.
Innerhalb des <OBJECT>-Tags werden die Attribute der RPC SINUMERIK-OCX hinterlegt.

Datei: Examples\Example3\Example3.html

```
<HTML>
  <HEAD>
        <TITLE>MCIS_RPC.OCX Example 3</TITLE>
  </HEAD>

  <BODY>

    <OBJECT classid=CLSID:EDF199C1-4F2E-11D3-9DC3-00A0249B4877 id=Machine1>
        <PARAM NAME="MachineID"              VALUE="M1">
        <PARAM NAME="MachineIP"              VALUE="195.208.2.233">
        <PARAM NAME="MachinePort"            VALUE="3011">
        <PARAM NAME="MachineTimeout"         VALUE="5">
        <PARAM NAME="HostID"                 VALUE="H1">
        <PARAM NAME="HostPort"               VALUE="3010">
    </OBJECT>

        <P align=center><STRONG>MCIS_RPC.OCX Example 3</STRONG>  </P>


        <TABLE border=2 align=center width=60% id=TABLE1>

                <TR>
                        <TD> R1 = </TD> <TD><LABEL id=R1Param></LABEL> </TD>
                </TR>

                <TR>
                        <TD> R5 = </TD> <TD><LABEL id=R5Param></LABEL> </TD>
                </TR>

        </TABLE>

  </BODY>
```

In der HTML-Seite sind folgende drei VBScript-Funktionen enthalten.

Window_OnLoad	Wird beim Laden der HTML-Seite aufgerufen.
Machine1_RxVARxH	Wird beim Eintreffen des RPC's R_VAR_H aufgerufen.
Machine1_RxREPORTxH	Wird beim Eintreffen des RPC's R_REPORT_H aufgerufen.

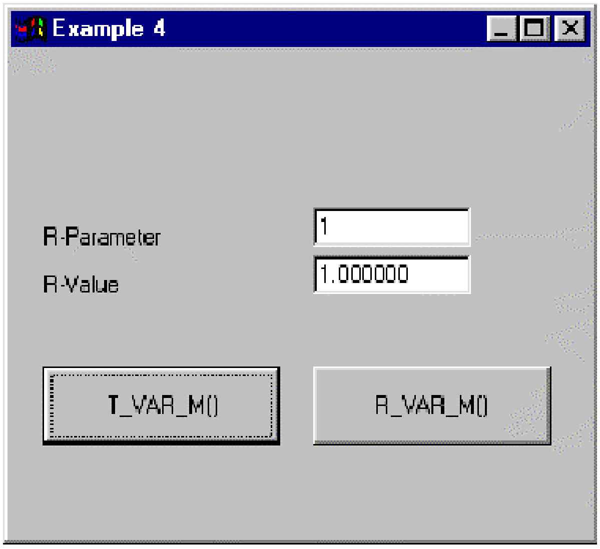
Fortsetzung der Datei: Examples\Example3\Example3.html

```
        <SCRIPT LANGUAGE="VBScript">

                Option Explicit

                Sub Window_OnLoad

                        dim ret

                        ret = Machine1.T_VAR_M(0, 0, "Set01", "")
                        if ret <> 0 then MsgBox "T_VAR_M()->" & ret


                End Sub

                Sub Machine1_RxVARxH( OrderNum, VarMode, VarSet, VarDescr, VarData )

                  dim pos

                  pos = InStr( VarData, "|" )

                  if pos = 0 then
                                R1Param.innerText = VarData
                        else
                                R1Param.innerText = Left(VarData, pos-1)
                                R5Param.innerText = Mid (VarData, pos+1)
                        end if

                End Sub

                Sub Machine1_RxREPORTxH( OrderNum, Typ, Number, Time, Flag, ResInt1,
                                                        ResInt2, ResByte )

                 If Number(0) <> 0 Then
                   MsgBox "On RPC R_REPEOR_H ( ... Number(0)->" & Number(0) & " )"
                 End If

                End Sub


        </SCRIPT>

</HTML>
```

10.5.4 Beispiel 4 - Lesen und Schreiben von R-Parametern (Visual J++)

In diesem Beispiel wird die gleiche Funktionalität wie im Beispiel 2 implementiert, jedoch unter Verwendung von MS Visual J++ 6.0 SP3.

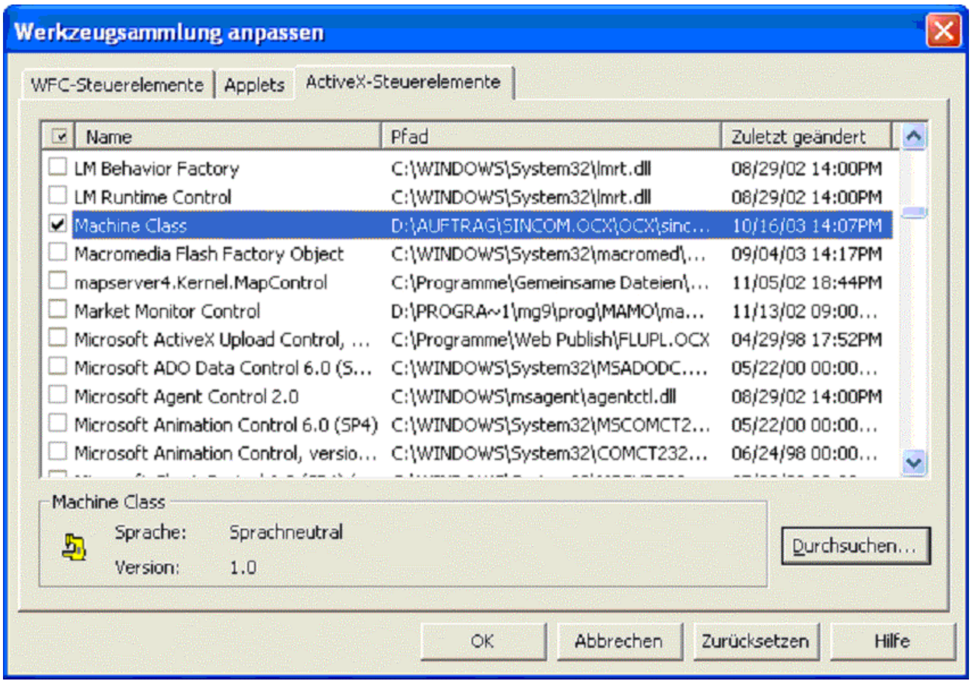
Das Ablaufschema entspricht dem Ablauf im Beispiel 2.



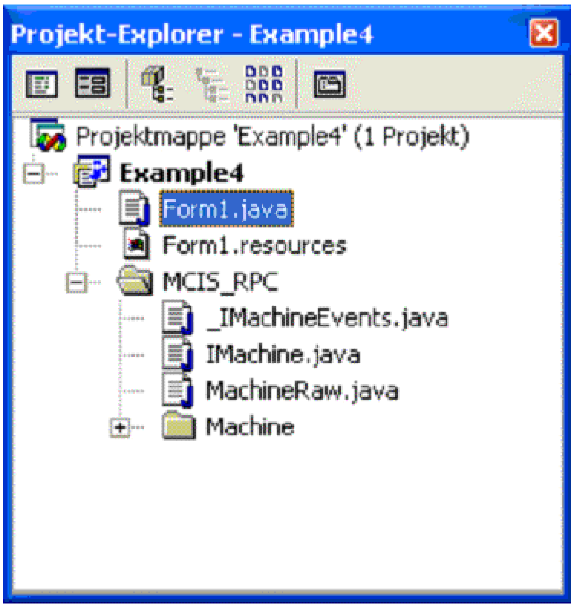
Einbindung der RPC SINUMERIK-OCX in MS Visual J++

Die Entwicklungsumgebung MS Visual J++ erlaubt es ActiveX Komponenten zu verwenden.
Die Einbindung erfolgt über das Menü: „**Tools -> Customize ToolBox -> ActiveX Controls**“

In dieser Maske muss der Eintrag „**Machine Class**“ ausgewählt werden.



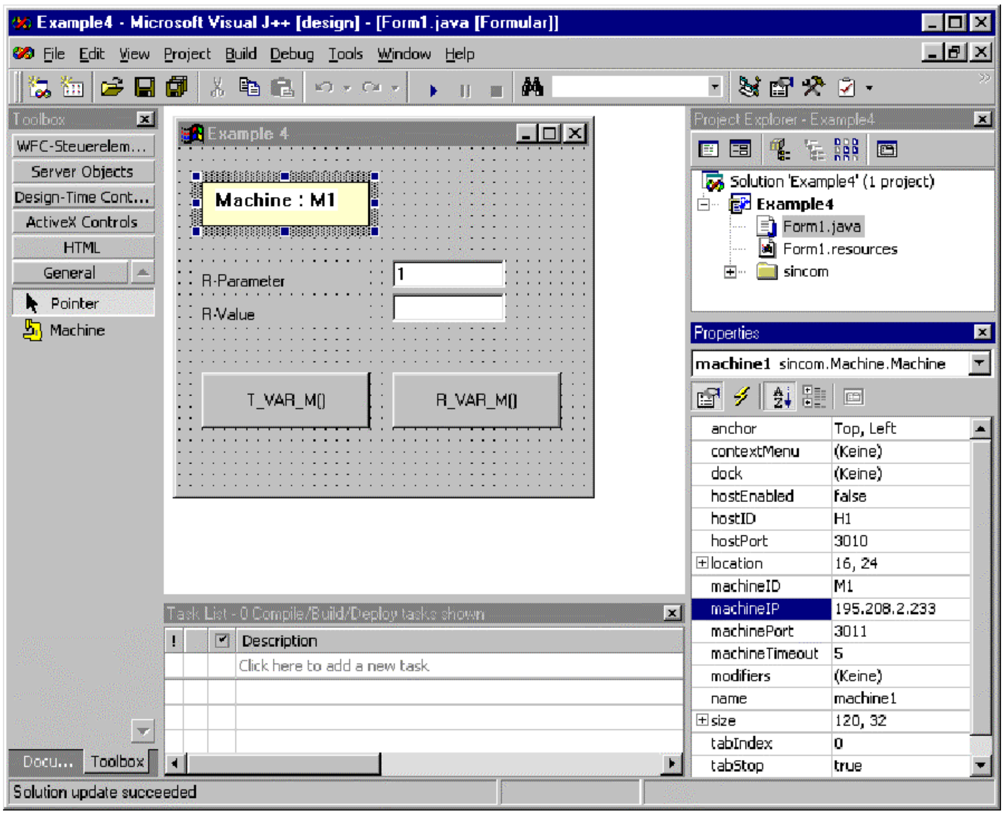
Mit der Einbindung der RPC SINUMERIK-OCX werden vom VJ++ zusätzliche Klassen im Verzeichnis RPC SINUMERIK generiert.


Projekt-Explorer - Example4
Projektmappe 'Example4' (1 Projekt)
Example4
Form1.java
Form1.resources
MCIS_RPC
_IMachineEvents.java
IMachine.java
MachineRaw.java
Machine

Source-Code der Visual-J++ Applikation

Über das Eingabefeld **R-Parameter** wird die Nummer des R-Parameters für das Lesen und Schreiben vorgegeben. In dem Eingabefeld **R-Value** wird - beim Lesen - der aktuelle Wert des R-Parameters angezeigt, bzw. - beim Schreiben - der neue Wert eingegeben.

Das Absenden der RPC's T_VAR_M() und R_VAR_M() erfolgt über die entsprechenden Schaltflächen.



Datei: Examples\Example4\Form1.java

```
private void cmdT_VAR_M_click(Object source, Event e)
{
        // read R parameter

        long      ret;
        String[]  VarDescr        = new String[1];          // item for access
        String[]  VarSet          = new String[1];

        VarSet[0]       = "";
        VarDescr[0]     = "/Channel/Parameter/R[" + txtRParam.getText() + "]";


        ret = machine1.T_VAR_M(0,0,VarSet,VarDescr );
        if ( ret != 0 ) MessageBox.show("T_VAR_M() -> " + ret);
}

private void cmdR_VAR_M_click(Object source, Event e)
{
        // write R parameter

        long      ret;
        String[]  VarDescr        = new String[1];          // item for access
        String[]  VarSet          = new String[1];
        String[]  VarData = new String[1];

        VarSet[0]       = "";
        VarDescr[0]     = "/Channel/Parameter/R[" + txtRParam.getText() + "]";
        VarData [0]     = txtRValue.getText();

        ret = machine1.R_VAR_M(0, 0, VarSet, VarDescr, VarData);
        if ( ret != 0 ) MessageBox.show("R_VAR_M() -> " + ret);
}

private void machine1_RxVARxH(Object source, MCIS_RPC.Machine.RxVARxHEvent e)
{
        // show R parameter in the form

        txtRValue.setText( e.VarData );
}

private void machine1_RxREPORTxH(Object source,
                                    MCIS_RPC.Machine.RxREPORTxHEvent e)
{
        int    ErrorNr = e.Number.getVariantArray()[0].getInt();

        if ( ErrorNr != 0 )
        {
           MessageBox.show( "On RPC R_REPORT_H ( ... Number(0)->" + ErrorNr + " )" );
        }
}
```

■

Für Notizen

I Index

I.1 Stichwortindex

■

Motion Control Infomation System

SINUMERIK 840D/840Di/810D
Rechnerkopplung RPC SINUMERIK

Nahtstelle zu PLC/NCK (FBR-NPL)

1

1 Nahtstelle RPC und Maschinen-PLC

1.1 Beschreibung

Beschreibung der Nahtstelle zwischen der Rechnerkopplungssoftware (RPC SINUMERIK) und der Maschinen-PLC.

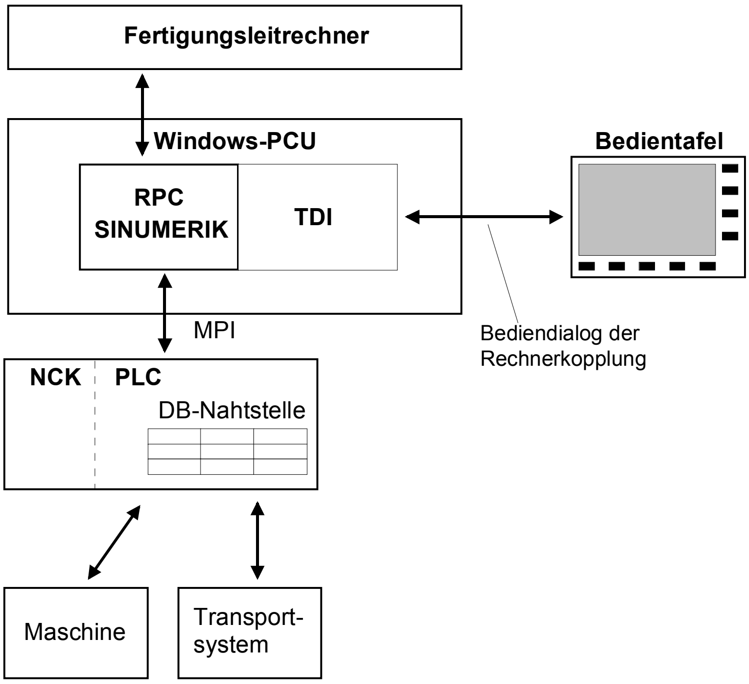


Bild 1-1 Übersicht

Hinweis

Verweise zum Teil 1 „Nahtstelle zum Fertigungsleitrechner“ werden in den folgenden Kapiteln notiert in der Form: (Siehe /NFL/ <Kapitel-Nr> Kapitelüberschrift), z.B. (Siehe /NFL/ Kap. 5.8 Übertragene Daten übernehmen)

Zur Kommunikation zwischen RPC SINUMERIK und PLC wird ein Nahtstellen-DB benötigt. Hierfür ist der Siemens-Standard-DB 12 reserviert. Der DB wird vom Anwender eingerichtet. Die Datenelemente der DB-Nahtstelle werden in Blöcken zusammengefasst, die jeweils einen Aspekt der Nahtstelle abdecken (z.B. Globaldaten, Haltestellendaten, NC-Programmzuordnung). Die Darstellung der einzelnen Blöcke erfolgt in Tabellenform. Alle Blöcke werden aneinanderfolgend im Nahtstellen-DB abgelegt.
Die binären Datenelemente vom Typ "int(WORD)" bzw. "Long(DWORD)" werden im S7-Format (Little-Endian) im DB abgelegt. Bei Zugriffen von der PLC erfolgt dort

eine entsprechende Wandlung in das Intel-Format (Big-Endian). Datenelemente, die einen Bezeichner darstellen, werden als Byte-Felder mit ASCII-Zeichen realisiert.
Die Beschreibung der Nahtstelle erfolgt in Tabellenform, in der Spalte „Zugriff“ steht, wer dieses Feld beschreibt. Hierbei werden folgende Kürzel verwendet:

- RPC SINUMERIK — Rechnerkopplungssoftware auf PCU (indirekt vom FLR)
- Bediener — Dialogprogramm der Rechnerkopplung
- PLC — Anwender PLC-Programm

Um den internen Aufwand für Kommunikation klein zu halten, wird jede Änderung in der Nahtstelle von RPC SINUMERIK mittels der "Anforderung von PLC" (Teil der Nahtstelle) mitgeteilt. RPC SINUMERIK nimmt das Eintreffen solcher Anforderungen sofort wahr. (Hotlink).

Maschine

Globaldaten	
1. Haltestelle	Die Anzahl der Haltestellen ist in den Globaldaten definiert
2. Haltestelle	
...	
n. Haltestelle	

Transportsystem

Globaldaten	
1. Haltestelle	Die Anzahl der Haltestellen ist in den Globaldaten definiert
2. Haltestelle	
...	
n. Haltestelle	
1. Transportauftrag	Die Anzahl der Transportaufträge ist in den Globaldaten definiert
2. Transportauftrag	
...	
n. Transportauftrag	

1.2 Globaldaten

Tabelle 1-1 Globaldatenliste

Datenelement	Kurzname	Daten-typ	Zugriff von	Offset
Anforderung von PLC	PLCReq	Byte	PLC/RPC SINUMERIK	0
Änderungs-Trigger	Trigger	Byte	PLC	1
Anforderung von RPC SINUMERIK	SCReq	Byte	RPC SINUMERIK /PLC	2
Maschinen-Modus	MODE_PLC	Byte	PLC	3
RPC-Modus	MODE_RPC	Byte	RPC SINUMERIK / Bediener	4
Kennung für WZ-Daten	DataTyp	Byte	PLC	5
Magazinnummer	MagNum	Word	PLC	6-7
Platznummer	PlaceNum	Word	PLC	8-9
T-Nummer	TNum	Word	PLC	10-11
Anzahl der Haltestellen	DockPosCount	Byte	PLC	12
Anzahl der Programm-zuordnungen	NC4WpcCount	Byte	PLC	13
Maschinenzustand	MachineStatus	Byte	PLC	14
NC-Betriebsart	MachineMode	Byte	PLC	15
Reserve 1	Reserve1	Word	PLC	16-17
Reserve 2	Reserve2	Word	PLC	18-19

Hinweis

Die einzelnen Einträge des Datenbausteins werden im Folgenden in der Tabellenreihenfolge erklärt.

Anforderung von PLC

Tabelle 1-2 Anforderung von PLC

Bit-Nr.	Funktion	Zugriff von
0	Werkstückträgerstatus geändert	PLC - 1/RPC SINUMERIK - 0
1	Werkzeug melden	PLC - 1/ RPC SINUMERIK - 0
2	Zustandsänderung	PLC - 1/ RPC SINUMERIK - 0

Die PLC zeigt mit diesem Byte Änderungen in der Nahtstelle an. Die PLC muss, nachdem sie das Anforderungsbyte gesetzt hat, im Triggerbyte das nächste Bit setzen (siehe unten).
Die PLC darf in das Anforderungsbyte erst dann wieder schreiben, wenn es von RPC SINUMERIK nach der Verarbeitung auf 0 gesetzt wurde.

Werkstückträgerstatus geändert

Werkstückträgerstatus geändert wird von der PLC gesetzt, wenn bei einem der Werkstückträger der Status von der PLC geändert wurde (Siehe Kap. 1.3 Haltestellendaten der Maschine) bzw. bei Palettenbewegungen innerhalb der Maschine.

Werkzeug melden

Werkzeug melden wird von der PLC gesetzt, wenn ein Werkzeug dem FLR gemeldet werden soll (z.B. WZ-Bruch). Welches Werkzeug zu melden ist, steht in den Datenelementen:
"Magazinnummer“ und
„Platznummer bzw. T-Nummer".

Zustandsänderung

Zustandsänderung wird von der PLC bei jeder Zustandsänderung (Maschinen-Modus , RPC-Modus, Maschinenzustand, NC-Betriebsart) gesetzt, über die der FLR informiert werden soll. RPC SINUMERIK muss daraufhin **R_MACHINE_H()** an den FLR senden. (z.B. Werkstückträgerankunft, NC-Start, NC-Ende, Modeumschaltungen etc.)

Änderungstrigger

Auf dieses Byte setzt die Rechnerkopplung einen DDE-Hotlink.
Die PLC setzt in diesem Byte jeweils ein Bit, wenn Änderungen von der PLC anstehen. Für jeden neuen Anstoß muss die PLC das nächste Bit setzten und das bisherige zurücksetzen, nach Bit 7 wird wieder mit Bit 0 begonnen.

Anforderung von RPC SINUMERIK

Tabelle 1-3 Anforderung von RPC SINUMERIK

Bit-Nr.	Funktion	Zugriff von
0	Synchronisationskennung	RPC SINUMERIK
1	Komponenten abschalten	RPC SINUMERIK
2	Komponenten einschalten	RPC SINUMERIK
3	Schreibzugriff auf Haltestellendaten anfordern	RPC SINUMERIK
4	Projektspezifische Sonderfunktion 1	RPC SINUMERIK
5	Projektspezifische Sonderfunktion 2	RPC SINUMERIK

Die einzelnen Bits werden von RPC SINUMERIK gesetzt und zurückgenommen.

Synchronisationskennung

Synchronisationskennung wird vom Leitrechner gesetzt und zurückgenommen. (Siehe /NFL/ Kap. 5.16.1 Synchronisation- Start/-Ende **C_SYNCH_M ()**) Der Zustand der Maschine muss für die Dauer der Synchronisation unverändert bleiben. Die PLC darf keine neue Bearbeitung starten und keine Palettenbewegungen durchführen.

Komponenten abschalten

Komponenten abschalten wird vom Leitrechner gesetzt (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M ()**). Es dient als Anforderung an die PLC Komponenten (Antriebe) abzuschalten (siehe auch Maschinenmodus, Komponenten abgeschaltet).

Komponenten einschalten

Komponenten einschalten wird vom Leitrechner gesetzt (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M ()**). Es dient als Anforderung an die PLC Komponenten (Antriebe) einzuschalten (siehe auch Maschinenmodus, Komponenten eingeschaltet).

Schreibzugriff auf Haltestellendaten anfordern

Schreibzugriff auf Haltestellendaten anfordern wird von RPC SINUMERIK als Anforderung gesetzt, wenn RPC SINUMERIK die Haltestellendaten (Werkstückträgerstatus, Folgebearbeitung, Seite der Bearbeitung) ändern will. Wenn die PLC den „Schreibzugriff auf Haltestellendaten erlaubt“ im Datenelement Maschinen Modus gesetzt hat, darf nur RPC SINUMERIK Änderungen durchführen. RPC SINUMERIK muss nach seiner Änderung die Anforderung zurücksetzen und danach muss dann auch die PLC das Flag „Schreibzugriff auf Haltestellendaten erlaubt“ zurücksetzen. Durch diese Koordination wird verhindert, dass RPC SINUMERIK auf Grund einer Palettenbewegung falsche Haltestellen beschreibt.

Projektspezifische Sonderfunktion1

Projektspezifische Sonderfunktion1 wird vom Leitrechner gesetzt. (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M 5**)
Projektspezifische Sonderfunktion1 wird vom Leitrechner zurückgesetzt. (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M 6**)

Projektspezifische Sonderfunktion2

Projektspezifische Sonderfunktion2 wird vom Leitrechner gesetzt. (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M 7**)
Projektspezifische Sonderfunktion2 wird vom Leitrechner zurückgesetzt. (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung **C_MODE_M 8**)

Maschinen-Modus

Tabelle 1-4 Maschinen-Modus

Bit-Nr.	Funktion	Zugriff von
1	Komponenten abgeschaltet (Feierabend)	PLC
2	Schreibzugriff auf Haltestellendaten erlaubt	PLC

Komponenten abgeschaltet

Komponenten abgeschaltet wird von der PLC gesetzt, wenn dieser Zustand erreicht ist. Die Anforderung dazu erfolgt durch Bit 1 in den Anforderungs-kennungen.

Schreibzugriff auf Haltestellendaten erlaubt

Schreibzugriff auf Haltestellendaten erlaubt wird von der PLC als Reaktion auf die Anforderung "Schreibzugriff auf Haltestellendaten anfordern" gesetzt. Durch diese Koordination wird verhindert, dass RPC SINUMERIK auf Grund einer Palettenbewegung falsche Haltestellen beschreibt.

RPC SINUMERIK-Modus

Tabelle 1-5 RPC SINUMERIK -Modus

Bit-Nr.	Betriebsart	Zugriff von
0	FLR-Modus unbemannt	Bediener
1	FLR-Modus bemannt	Bediener
2	Manueller-Modus	Bediener
3	Sondermodus	RPC SINUMERIK, Bediener
4	Offline FLR1	RPC SINUMERIK
5	Offline FLR2	RPC SINUMERIK
6	Wird bei R_MACHINE_H als Machine Mode 500 gemeldet	PLC
7	Wird bei R_MACHINE_H als Machine Mode 600 gemeldet	PLC

Der RPC SINUMERIK Modus kann über einen RPC SINUMERIK Dialog vom Bediener gesetzt werden. Der Sondermodus kann auch durch **C_MODE_M ()** vom FLR an- und abgewählt werden. Wenn RPC SINUMERIK eine Verbindungsunter-brechung feststellt, setzt sie das Bit für "Offline". Wenn das Offline-Bit gesetzt ist, werden von RPC SINUMERIK keine Daten mehr an den FLR gesendet.

FLR bemannt/unbemannt

In den FLR-Modi **bemannt** und **unbemannt** erfolgt der NC-Start von PLC (auf Initiative vom FLR), der Unterschied zwischen **FLR-Modus unbemannt** und **FLR-Modus bemannt** liegt darin, dass bei unbemannter Produktion eventuell andere Strategien bei Störungen ausgeführt werden sollen als bei bemannter Produktion.

Manueller Modus

Bei **Manueller Modus** erfolgt kein automatischer NC-Start, die Maschine unterliegt jedoch dem automatischen Materialfluss.

Sondermodus

Im **Sondermodus** erfolgt kein automatischer NC-Start und die Maschine unterliegt auch nicht dem automatischen Materialfluss.

Offline

Offline bedeutet, dass keine Verbindung zum FLR besteht, es werden keine Daten an den FLR gesendet. **Offline** wird aufgehoben, wenn RPC SINUMERIK durch einen vom FLR ankommenden RPC erkennt, dass die Verbindung wieder besteht.

Kennung für WZ-Daten

Über die **Kennung für WZ-Daten** kann einer von drei WZ-Datensätzen angewählt werden, der an den FLR übertragen wird. Welche Datenbereiche in diesen Sätzen enthalten sind, wird über das Konfigurationsprogramm festgelegt. Es sind die Kennungen 21, 22, 23 erlaubt. Die Kennung wird mit den WZ-Daten zum FLR übertragen (Siehe /NFL/ Kap. 4.1 Werkzeigdaten).

Magazinnummer, Platznummer, T-Nummer

Das zu meldende Werkzeug wird durch Magazinnummer und Platznummer oder alternativ durch T-Nummer vorgegeben. Bei Vorgabe der **T-Nummer** ist **Magazinnummer** und **Platznummer** auf 0 zu setzen und umgekehrt.
Mit der Anforderung von PLC: **Werkzeug melden** wird RPC SINUMERIK aufgefordert, die Datenelemente

- Magazinnummer,
- Platznummer,
- T-Nummer und
- Kennung für WZ-Daten

zu lesen, und die Werkzeugdaten an den FLR zu übertragen. Im Anschluss muss RPC SINUMERIK diese Datenelemente löschen (mit Nullen füllen).

Hinweis

WZ-Meldungen die im Rahmen von WZ-Beladen und WZ-Endladen auftreten, werden nicht vom PLC-Anwenderprogramm initiiert (siehe hierzu /NFL/).

Anzahl der Haltestellen

Die Anzahl der Haltestellen der Maschine wird bei der Inbetriebnahme der Maschine statisch hinterlegt. Sie entspricht der Anzahl der Haltestellendaten-Blöcke in der Nahtstelle (siehe Kap.1.3 Haltestellendaten der Maschine).

Anzahl der Programm-zuordnungen

Anzahl der Programzuordnungen wird bei der Inbetriebnahme der Maschine statisch hinterlegt. Sie entspricht der Anzahl der NC-Programmzuordnung -Blöcke in der Nahtstelle (siehe Kap.1.4 NC-Programmzuordnung).

Maschinenzustand

Tabelle 1-6 Maschinenzustand

Bit-Nr.	Funktion	Zugriff von
0	Maschine ist tätigt	PLC
1	Maschine ist gestört	PLC
2	Neuanlauf der Maschine	PLC

NC-Betriebsart

Tabelle 1-7 NC-Betriebsart

Bit-Nr.	Funktion	Zugriff von
0	Automatic	PLC
1	MDA	PLC
2	JOG	PLC
3	TEACH IN	PLC

Der Maschinenzustand und die NC-Betriebsart werden an den FLR mit **R_MACHINE_H** gemeldet. Sie werden jedoch nicht im RK-Server ausgewertet.

Reserve 1 und 2

Diese Variablen können vom Maschinenhersteller mittels der PLC beliebig benutzt werden. Die Werte werden bei **R_MACHINE_H ()** an den FLR gemeldet, wobei üblicherweise im FLR keine Verarbeitung für diese Werte stattfindet.

1.3 Haltestelledaten der Maschine

Die Haltestellendaten beschreiben jeweils einen Maschinenplatz (Bearbeitungsplatz, In/Out-Platz). Die Anzahl der Maschinenplätze ist im Datenelement **Anzahl der Haltestellen** der Globalendaten hinterlegt.

Tabelle 1-8 Haltestellendaten

Datenelement	Kurzname	Datentyp	Zugriff von
Haltestellenstatus	DockPos Status	Byte	PLC
Werkstückträgerstatus	WPCStatus	Byte	PLC/RPC SINUMERIK
Werkstückträger	WPC	Byte[6]	PLC
Seite der Bearbeitung	ClampCube Side	Word	PLC/ RPC SINUMERIK
Folgebearbeitung	FB	Byte	RPC SINUMERIK
Reserve1	Reserve1	Byte	PLC

Haltestellenstatus

- Bit 0 = Gestört
- Bit 1 = Gesperrt

Das Bitfeld beschreibt den aktuellen Zustand der Haltestelle, es wird von der PLC gesetzt. Ist kein Bit gesetzt ist die Haltestelle freigegeben. Das Bit **Gestört** wird auf Grund von Peripherie-Signalen gesetzt bzw. zurückgenommen. Die Störungs-ursache wird über die Funktion Meldungen (Siehe /NFL/ Kap. 5.5 Meldungen **R_REPORT**) dem Leitrechner mitgeteilt. Die PLC führt keine Palettentransporte zwischen Plätzen die **Gestört** sind aus. Ist die Haltestelle **Gesperrt**, darf sie nicht vom Transportsystem angefahren werden.

Werkstückträgerstatus

Tabelle 1-9 Werkstückträgerstatus

Bit-Nr.	Funktion	Zugriff von
0	neu angeliefert (ohne Programmzuordnung)	PLC
1	Bearbeitung ist vorgesehen (mit Programmzuordnung)	RPC SINUMERIK
2	Programmanwahl vorbereiten	PLC
3	Programmanwahl erreicht	RPC SINUMERIK
4	In Bearbeitung	PLC
5	Bearbeitung beendet	PLC
6	Bearbeitung abgebrochen	PLC
7	Bearbeitung ist nicht vorgesehen (nur zur Pufferung)	PLC

Neu angeliefert

Der Status **neu angeliefert** wird von der PLC für einen neu angelieferten Werkstückträger vergeben. (Ausnahme: Bearbeitung ist nicht vorgesehen). Dieser Status veranlasst den Leitrechner die Programmzuordnung durchzuführen. Ist die Programmzuordnung erfolgt, wird der Status "Bearbeitung ist vorgesehen" von dem RK-Server gesetzt.

Bearbeitung ist vorgesehen

Für Werkstückträger mit dem Status **Bearbeitung ist vorgesehen** wird, sobald die laufende Bearbeitung beendet ist, von der PLC der Status **Programmanwahl vorbereiten** gesetzt.

Programmanwahl

Nach erfolgter Programmanwahl - das vom Leitrechner für diese Palette zugeordnete Programm wurde in NCK geladen und für die Bearbeitung ausgewählt - setzt der RK-Server den Status **Programmanwahl** erreicht für den jeweiligen Werkstückträger. Die PLC kann nun den NC-Start auslösen. Es liegt jedoch in der Verantwortung des Maschinenherstellers, dass ein NC-Start nur dann von der PLC erfolgt, wenn alle sicherheitsrelevanten Aspekte erfüllt sind (z.B. Schutztür zu usw.).

In Bearbeitung

Nach dem die Bearbeitung gestartet wurde, setzt die PLC den Status **In Bearbeitung**. Wurde die Bearbeitung beendet, dann setzt die PLC den Status **Bearbeitung beendet** für den entsprechenden Werkstückträger. Werkstückträger mit dem Status **Bearbeitung beendet** werden von der PLC selbstständig auf einen Endladeplatz transportiert.

Bearbeitung beendet

Ist die Kennung **Folgebearbeitung** gesetzt, verbleibt der Werkstückträger auf dem Bearbeitungsplatz. Der RK-Server setzt, als Reaktion auf den Status **Bearbeitung beendet**, erneut den Status **Bearbeitung ist vorgesehen**. Als Reaktion darauf fordert die PLC beim RK-Server - mit dem Status **Programmanwahl vorbereiten** - erneut die Durchführung der Programmanwahl für eine Folgebearbeitung an. Der weitere Ablauf entspricht dem bei der Erstbearbeitung.

Bearbeitung abgebrochen

Der Status **Bearbeitung abgebrochen** wird gesetzt, wenn nach einer Störung ein Werkstückträger nicht weiter bearbeitet wird. Diese Kennung dürfte vor allem bei unbemannter Produktion auftreten. Ein Werkstückträger mit dieser Kennung darf nicht für einen weiteren Arbeitsgang auf eine andere Maschine gebracht werden, er darf nur auf einen Ablageplatz gebracht werden.

Bearbeitung nicht vorgesehen

Ein Werkstückträger, der zur Pufferung von dem TPS angeliefert wurde, erhält nicht den Status **neu angeliefert** sondern den Status **Bearbeitung ist nicht vorgesehen**. Diese Information übergibt das TPS an die PLC. Für Werkstückträger mit diesem Status erfolgt keine Programmzuordnung durch den FLR.

Werkstückträger

Bezeichner des Werkstückträgers der sich aktuell auf der Haltestelle befindet (z.B. „WST01“). Diese Information wird von der PLC eingetragen. Dies setzt voraus, dass diese Information vom Transportsystem bzw. direkt vom Werkstückträger übernommen werden kann. Ist kein Werkstückträger auf der Haltestelle muss das Feld mit binär 0 gefüllt werden.
Der Bezeichner muss mit ‘\0‘ hinter dem letzten Zeichen abgeschlossen werden, da RPC SINUMERIK einen String erwartet, es sind also maximal 5 Byte lange Bezeichner zulässig.

Folgebearbeitung

Diese Kennung wird gleichzeitig mit dem Bearbeitungsstatus **Programmanwahl erreicht** vom RK-Server gesetzt. Sie informiert die PLC, ob außer der aktuellen Bearbeitung eine Folgebearbeitung vorgesehen ist. Die PLC nutzt diese Information, um den Werkstückträgertransport innerhalb der Maschine zu steuern.

Seite der Bearbeitung

Diese Information wird gleichzeitig mit dem Bearbeitungsstatus **Programmanwahl erreicht** vom RK-Server gesetzt. Die PLC benutzt diesen Wert, um eine entsprechende Seite eines Spannwürfels für die Bearbeitung einzustellen, bzw. gibt diesen Wert an NCK weiter.
Die Rechnerkopplung gibt bei Spannwürfeln die Reihenfolge der Seiten der Bearbeitung anhand der Programmzuordnung vor. Wird eine Beeinflussung der Bearbeitungsreihenfolge von der PLC gewünscht, dann können optional, über ein Eintrag in der Konfigurationsdatei des RK-Servers, die Programmzuordnungsdaten in einem separaten Datenbaustein der PLC gespiegelt werden. Die PLC erhält somit einen Lesezugriff auf diese Daten. Die von der PLC getroffene Auswahl der Bearbeitungsseite wird dem RK-Server im Feld **Seite der Bearbeitung** mitgeteilt. Dies erfolgt gleichzeitig mit dem Bearbeitungsstatus **Programmanwahl vorbereiten**. Der RK-Server führt die Programmanwahl für die von der PLC vorgegeben Seite durch. Der übrige Ablauf bleibt unverändert.

1.4 NC-Programmzuordnung (Option)

Diese Nahtstelle ist optional und kommt zum Einsatz, wenn bei Werkstückträgern mit Spannwürfel, die Reihenfolge der Bearbeitung der einzelnen Seiten durch die PLC vorgegeben werden soll und nicht durch den FLR. Es handelt sich dabei um Daten, die von RPC SINUMERIK verwaltet werden und hier zum lesen für die PLC gespiegelt werden.
Die Anzahl der NC-Programmzuordnungsblöcke ist im Datenelement **Anzahl der Programmzuordnungen** der Globaldaten (Kap. 1.3 Haltestelledaten der Maschine) hinterlegt.

Tabelle 1-10 NC Programmzuordnung

Datenelement	Kurzname	Datentyp	Zugriff von
Werkstückträger	WPC	Byte[6]	RPC SINUMERIK
Seite der Bearbeitung	ClampCubeSide	Word	RPC SINUMERIK
NC-Programmkennung	NCProgrammMark	Byte	RPC SINUMERIK
Bearbeitungsstatus	Status	Byte	RPC SINUMERIK
NC-Programm	NC-Programm	Byte[128]	RPC SINUMERIK

Werkstückträger, Seite der Bearbeitung

Mit der NC-Programmzuordnung wird einem "Werkstückträger" und einer "Spannwürfelseite (Seite der Bearbeitung)" ein NC-Programm zugeordnet.

NC-Programmkennung

Die NC-Programmkennung dient zur Erkennung, ob auf mehreren Seiten eines Spannwürfels das selbe NC-Programm benutzt wird. Beim Eintrag durch RPC SINUMERIK wird für die erste Seite eines Werkstückträgers "NCProgrammMark" = 1 gesetzt. Hat die nächste Seite ein anderes NC-Programm, wird "NCProgrammMark" = 2 gesetzt usw. Hätte nun die dritte Seite das selbe NC-Programm wie Seite 1, wird dessen "NCProgrammMark" übernommen. Damit besteht die Möglichkeit die Abarbeitungsreihenfolge so zu steuern, dass Seiten mit gleichem NC-Programm nacheinander bearbeitet werden.
Spannwürfelseiten, die mit gleichen NC-Programmen bearbeitet werden, erhalten auch gleiche NC-Programmkennung.

Bearbeitungsstatus

- Bit 1 = Bearbeitung ist vorgesehen
- Bit 4 = In Bearbeitung
- Bit 5 = Bearbeitung beendet

Listengröße

Die Größe der Liste zur NC-Programmzuordnung ist konfigurierbar, daraus ergibt sich die benötigte Länge im DB. Sie errechnet sich aus Anzahl der Haltestellen (Liegeplätze) der Maschine mal der maximalen Anzahl der nutzbaren Seiten eines Spannwürfels.

■

Für Notizen

2

2 Abläufe an der DB-Nahtstelle

Beschreibung

Die Kommunikation findet über die oben beschriebenen Nahtstellen-DB´s statt. Wer wann in welche Felder der Nahtstelle schreibt oder daraus liest, wird im Folgenden beschrieben.

2.1 Werkstückträgerankunft

Bei Ankunft eines Werkstückträgers an einer Haltestelle der Maschine muss die PLC entweder die Bezeichnung des Werkstückträgers lesen oder das Transportsystem muss sie der Maschine übergeben. Zusätzlich muss das Transportsystem der Maschine mitteilen (falls diese Information nicht vom FLR an die Maschine gemeldet wird [NC4WPC_M]), ob der Werkstückträger zur Bearbeitung oder nur zur Pufferung angeliefert wird. Wie diese Informationen zwischen TPS und Maschine übertragen werden, ist nicht Bestandteil dieser Beschreibung.
Die PLC muss den Haltestellen-Status sowie Werstückträger-Nummer und -Status in den entsprechenden Nahtstellen-DB schreiben und danach die Anforderungs-kennung Zustandsänderung in der Nahtstelle für Zustände setzen und den Änderungstrigger hochsetzen. RPC SINUMERIK sendet daraufhin **R_MACHINE_H ()** an den FLR.

2.2 Fertigungsdialog zwischen PLC/NCK und RPC SINUMERIK

Hat der FLR anhand von **R_MACHINE_H ()** erkannt, dass ein zu bearbeitender Werkstückträger an der Maschine angekommen ist, sendet er eine oder mehrere NC-Programmzuordnungen **NC4WPC_M ()**.RPC SINUMERIK trägt diese Informationen in eine interne Liste und optional in die Nahtstelle für NC-Programmzuordnung ein.

Daten für R_MACHINE ermitteln

Es müssen die Nahtstellen für Zustände und Haltestellendaten gelesen werden sowie die SINUMERIK-Betriebsart (Automatik, MDA, JOG) und der Programm-zustand. Die Betriebsart, die dem FLR gemeldet wird, setzt sich aus der Rechnerkopplungs-Betriebsart und der SINUMERIK-Betriebsart zusammen.

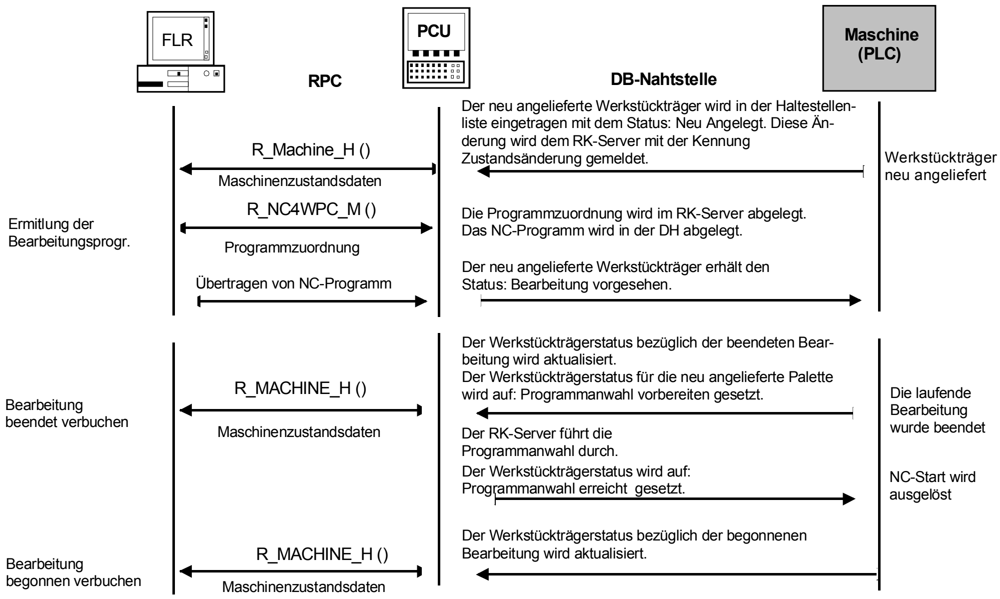


Bild 2-1 Ablauf Schema

■

Für Notizen

3

3 Dialogprogramm zu RPC SINUMERIK

3.1 Dialogprogramm von RPC SINUMERIK

Das Dialogprogramm von RPC SINUMERIK ist als eigenständige Bereichsapplikation im Bedienbaum der HMI-Advanced verfügbar.

Das Dialogprogramm bietet folgende Funktionen an:

- Zustand von RPC SINUMERIK anwählen
- Anfordern von NC-Programmen vom FLR
- Übertragen von NC-Programmen an den FLR
- Umschaltung zu MCIS-TDI (BE- / Entladen von Werkzeugen zum FLR)

3.2 Zustand von RPC SINUMERIK

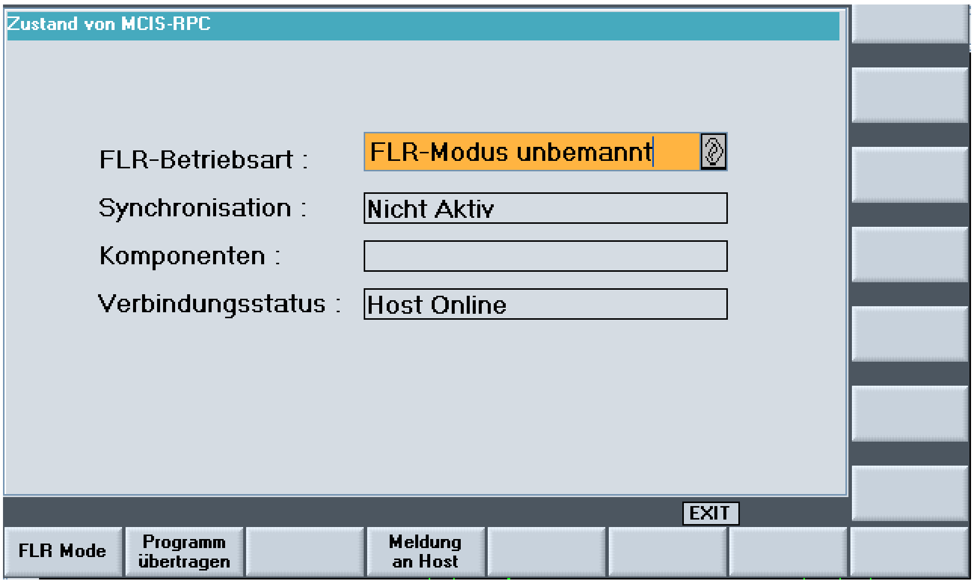


Bild 3-1 Die Maske ermöglicht das Ändern der FLR–Betriebsart

Modi von RPC SINUMERIK

- FLR-Modus unbemannt
- FLR-Modus bemannt
- Manueller-Modus
- Sondermodus

Es werden zusätzlich für RPC SINUMERIK relevanten Zustände angezeigt.

3.3 Programm übertragen

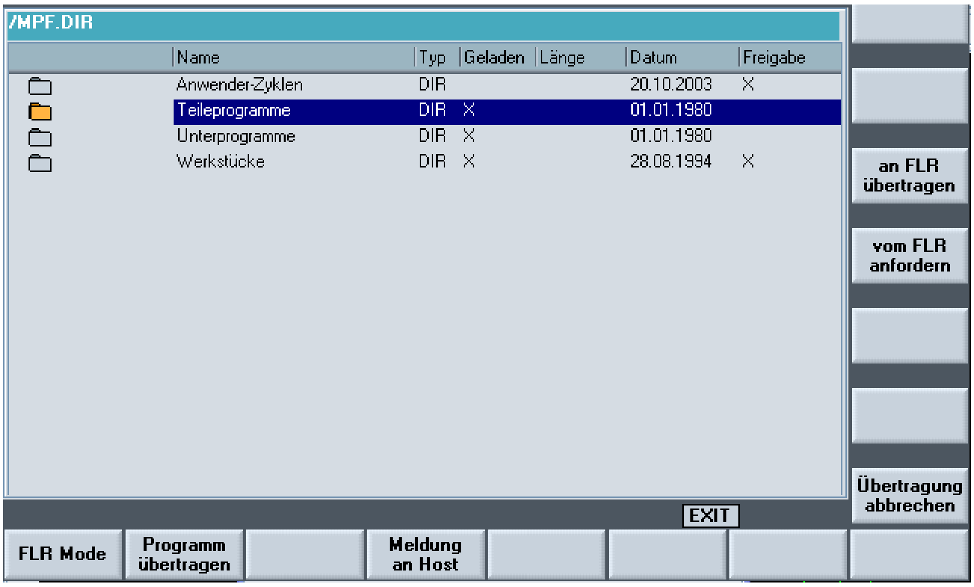


Bild 3-2 NC–Programme/ Dateien übertragen

Mit dem Softkey „an FLR übertragen" kann der Bediener Dateien aus dem angebotenem Datenhaltungsschema des HMI-Advanced an den Fertigungsleitrechner übertragen bzw. mit dem Softkey „vom FLR anfordern" zum HMI-Advanced übertragen.

3.3.1 Programm zum FLR senden

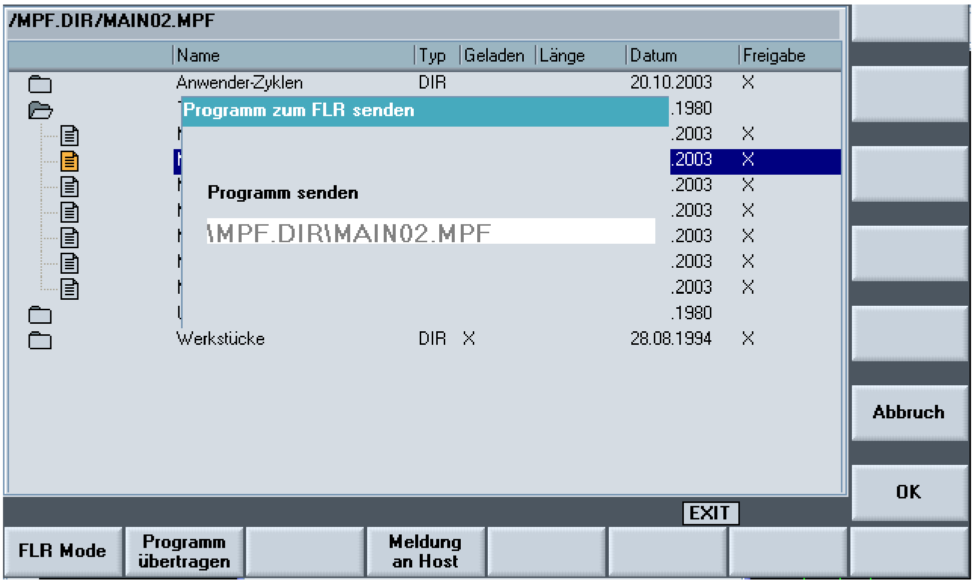


Bild 3-3 Programm zum FLR senden

3.3.2 Programm vom FLR anfordern

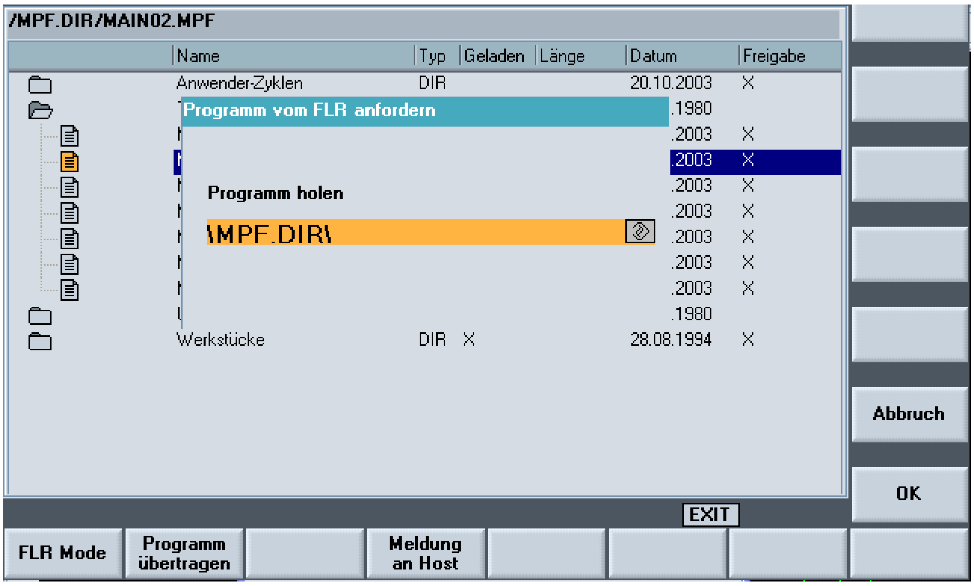


Bild 3-4 Anfordern eines Programmes vom FLR

Die Maske ermöglicht dem Bediener die Eingabe des NC–Programmnamens, das vom FLR angefordert werden soll.

3.3.3 Meldung an Host senden

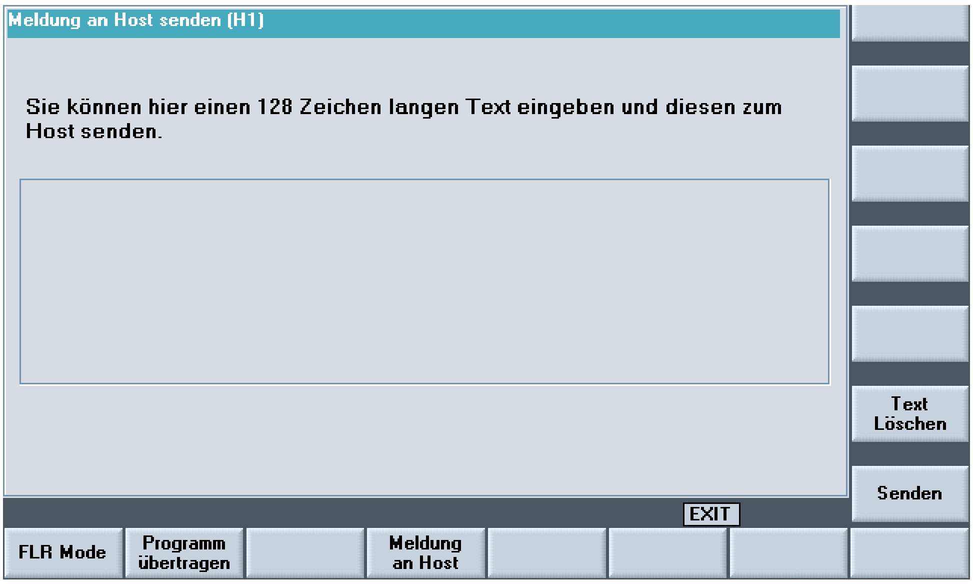


Bild 3-5 Meldung an Host senden

In die Textbox kann ein 128 Zeichen langer Text eingegeben werden. Folgende vertikale Softkeys stehen zur Verfügung oder werden bei Bedarf eingeblendet.

- Host Meldung quittieren
- Host wählen
- Text löschen
- Senden

Host Meldung quittieren

Wurde eine Meldung vom Host an den MMC103/PCU50 gesendet, so wird dieser als Alarm 65535 mit dem übertragenen Meldungstext eingetragen. Über den Softkey „Host-Meldung quittieren" kann der anstehende Alarm gelöscht werden.

Host wählen

Die Schaltfläche ist nur sichtbar, wenn mehr als ein Host konfiguriert wurde. Es erscheint ein Auswahlfeld über das der Host selektiert werden kann, an den die Meldung gesendet werden soll.

Hinweis

Default-Host ist der erste konfigurierte Host.

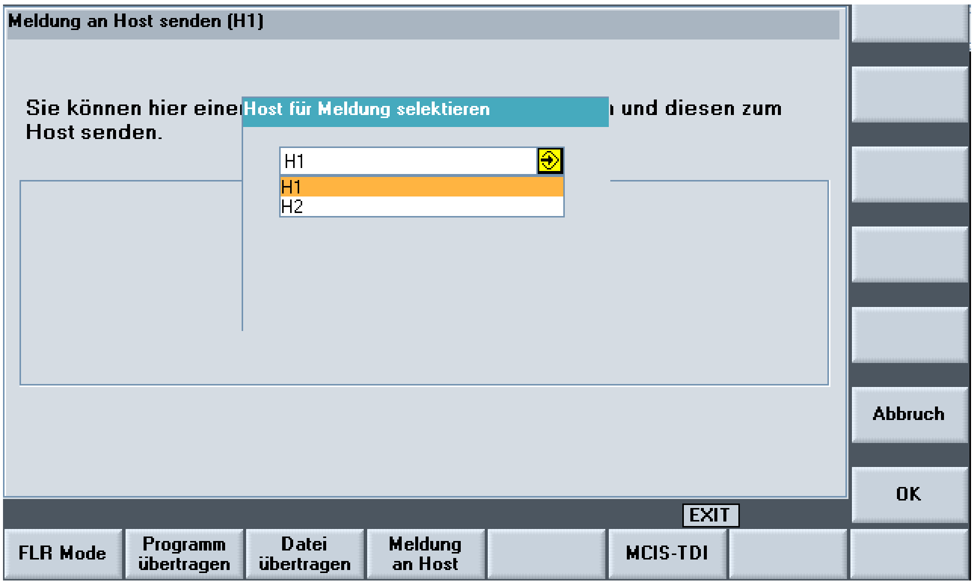


Bild 3-6 Host wählen

Text löschen

Der eingegebene Text wird gelöscht.

Senden

Der eingegebene Text wird an den Host gesendet.

3.3.4 Be-/Entladen von Werkzeugen zum FLR

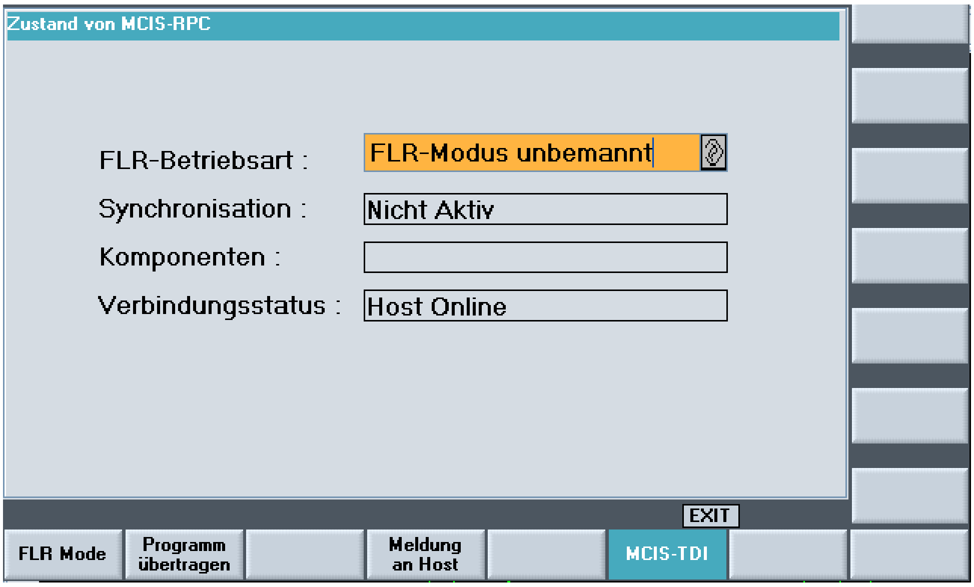


Bild 3-7 Maske mit MCIS-TDI Schaltfläche

MCIS-TDI

Die Schaltfläche ist nur sichtbar, wenn MCIS-TDI Toolhandling (Tool Data Information) auf dem HMI-Advanced installiert ist.

3.3.5 MCIS-TDI (Tool Data Information) Toolhandling

Allgemeines

Mit dem Modul MCIS-TDI Toolhandling können sämtliche Werkzeughantierungen innerhalb einer Anlage abgebildet, die Be- / Entlade- und Umsetzvorgänge auf der CNC-Maschine bzw. einem Werkzeuglager durchgeführt und die Werkzeugdaten von einem z.B. Fertigungsleitrechner übernommen werden. Bei den Vorgängen können beliebige Container innerhalb einer Anlage beteiligt werden. Das Modul TDI Toolhandling läuft, sowohl im vernetzten Verbund mit TDICell/Plant (Optional), als auch auf einem Einzelplatz (z.B. für WZ-Datenübernahme von einem Fertigungsleitrechner).

Hinweis

MCIS-TDI ist nicht im Lieferumfang von RPC SINUMERIK und muss gesondert bestellt werden.

Werkzeughantierung/Auswahl

Über die Einstiegsmaske der Werkzeughantierung können die zu bewegenden Werkzeuge ausgewählt werden.
Die Maske Werkzeughantierung/Auswahl ist zweigeteilt. Im oberen Teil werden im Explorer alle der lokalen Komponente (Bedien-/Installationsort) überlagerten Werkzeugmagazine innerhalb der Anlagenhierarchie angezeigt. Im unteren Teil ist die lokale Anlagenkomponente dargestellt

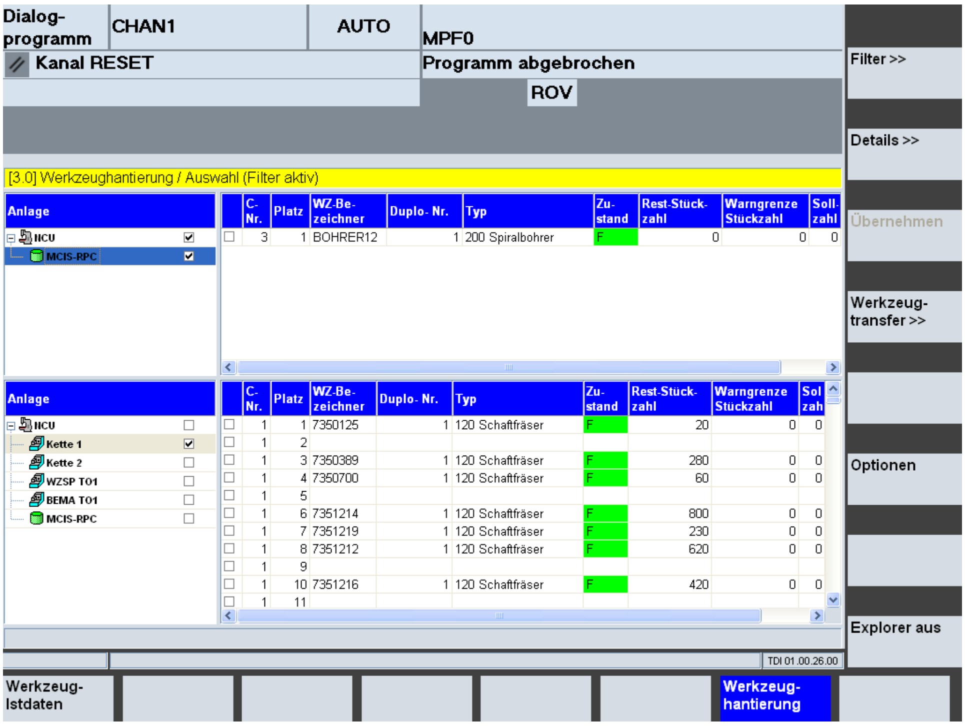


Bild 3-8 Werkzeughantierung/Auswahl

Hinweis

Im unteren Teil :
In der CNC-Version wird immer die Sicht auf die lokale Komponente (Einheit) angezeigt.
In der PC-Version wird die Sichtweise auf die Gesamtanlage, wie in der Konfiguration eingestellt, angezeigt.

In Abhängigkeit vom Bewegungsvorgang (Be- oder Entladen) kann sowohl die obere als auch die untere Anzeige Quelle oder Ziel der Bewegung sein.

Durch Auswahl bestimmter Werkzeuge in der Tabelle (setzen von Häkchen in der Auswahlbox) werden diese für die Bewegung bestimmt.
Die so ausgewählten Werkzeuge werden durch den Softkey „Übernehmen“ in die Bewegungsliste übertragen.

Es können sowohl einzelne als auch mehrere Werkzeuge (z.B. Kassette, WZ-Wagen, Belade-Magazin mit mehreren Plätzen) für die Bewegungsliste ausgewählt werden.
Wenn nur ein Werkzeug angewählt wurde (Quelle), kann im Ziel ein bestimmter Platz durch setzen des Häkchens in der Tabelle vorgegeben werden. Dies funktioniert ausschließlich bei Einzelwerkzeugen.

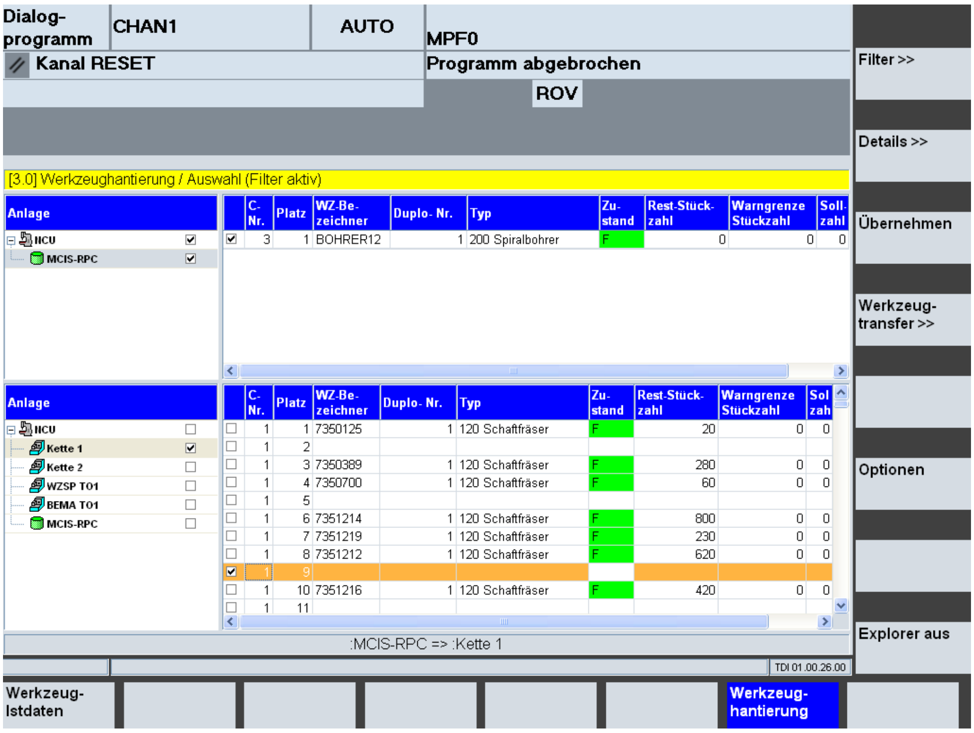


Bild 3-9 Werkzeughantierung mit Infozeile

Die Quelle und das Ziel der aktuellen Bewegungszusammenstellung werden explizit in der Infozeile dargestellt. Dies erfolgt unter Angabe von:

- Ort
- Container

Hinweis

- Bei einer Werkzeugbewegung darf maximal eine Maschine als Ziel beteiligt sein.
- Sind Werkzeuge durch den Vorgang „Übernehmen“ in die Bewegungsliste übernommen worden, so können sie nicht mehr angewählt werden und die Häkchen werden optisch anders gekennzeichnet (gegraut). Änderungen können nur in der Maske “Bewegungen organisieren" vorgenommen werden.
- Wird die Auswahl der anzuzeigenden Container im Explorer verändert, so verliert die Werkzeugauswahl des abgewählten Containers seine Gültigkeit. Dies gilt so lange die angewählten Werkzeuge noch nicht in die Bewegungsliste übernommen worden sind.
- Die Häkchen der angewählten Werkzeuge, die in die Bewegungsliste übernommen wurden, bleiben solange gesetzt, bis die Hantierung abgeschlossen ist.
- Die Bewegungsliste kann jederzeit durch „Werkzeug übernehmen“ erweitert werden.
- Eines der möglichen Hantierungsziele ist der Demontagecontainer, der in der Anlagenkonfiguration (Explorer) mit dem Symbol Mülleimer dargestellt wird. Werkzeuge die dorthin bewegt werden, bleiben nur eine begrenzte Zeit in der Liste des Containers gespeichert. Diese Liste kann einem übergeordneten Werkzeugmanagementsystem für eine Demontagefunktionalität zur Verfügung gestellt werden.

Softkeybeschreibung

Folgende Funktionen können über die vertikalen Softkeys ausgeführt werden:

Filter
Hierüber öffnet sich die Maske zur Definition von Selektionsfiltern, sowie zur Auswahl der in der Liste dargestellten Spalten. Der hier vorgenommene Filtereinstellung wirkt nur auf den Bereich der Maske auf der der Cursor steht.

Details
Es werden die Details des selektierten Werkzeuges zum Einsehen und Ändern von Werkzeugdaten aufgeblendet, wobei zwischen NC-Daten und PLC-Daten unterschieden wird.

Übernehmen
Die angewählten Werkzeugbewegungen werden in die Werkzeugbewegungsliste übernommen.

Werkzeug-Bewegung
Hierüber öffnet sich die Liste aller aktuellen Werkzeugbewegungen.

Optionen
Aufblenden der Pop-Up Maske zur Vorgabe unterschiedlicher Bewegungsabläufe für die Werkzeugbewegung.

Explorer ein/aus
Die Baumdarstellung der Anlagenkomponente wird ein- und ausgeblendet.

Für weitere Informationen siehe Funktionsbeschreibung MCIS-TDI
Ausgabe: 07.03
Bestell-Nr.: 6FC5297-6AE01-0AP0

3.3.6 Be-/Entladen von Werkzeugen zum FLR mit TDI

Allgemein

Über die Einstiegsmaske der Werkzeughantierung können die zu bewegenden Werkzeuge ausgewählt werden.
Die Maske Werkzeughantierung/Auswahl ist zweigeteilt. Im oberen Teil werden im Explorer alle der lokalen Komponente (Bedien-/Installationsort) überlagerten Werkzeugmagazine innerhalb der Anlagenhierarchie angezeigt (Default Einstellung). Im unteren Teil ist die lokale Anlagenkomponente dargestellt.

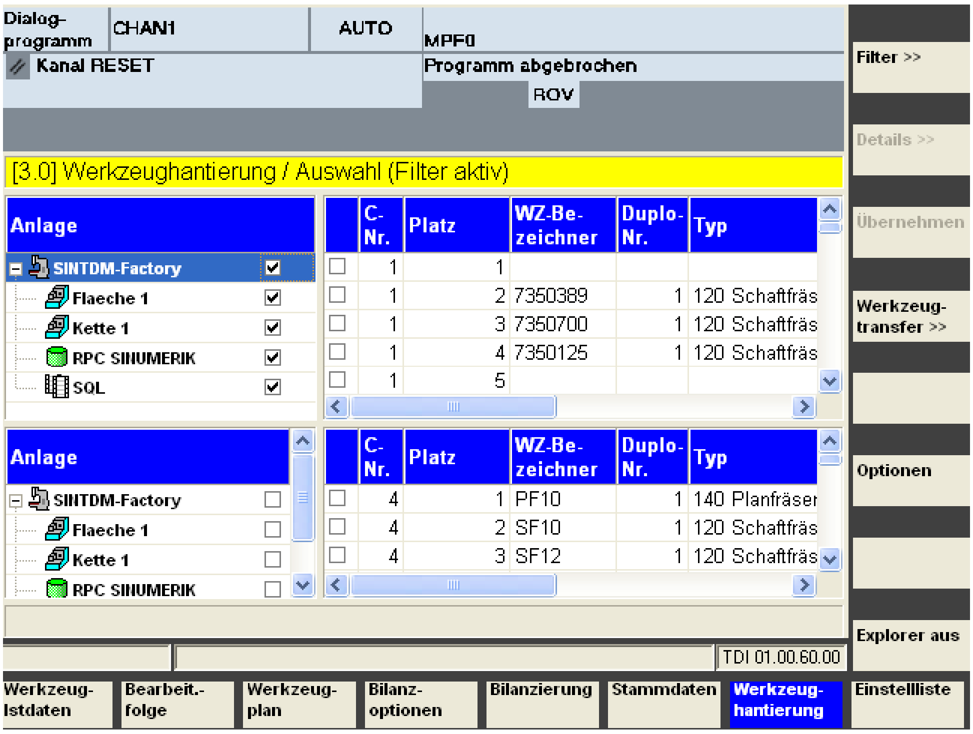


Bild 3-10 Einstiegsmaske Werkzeughantierung/Auswahl

Anfordern von Werkzeugdaten vom FLR

Damit Werkzeugdaten vom FLR angefordert werden, muss in der oberen Explorersicht der RPC-Container angewählt werden.
Über die Filtereinstellungen wird in der Maske Filter mit „Anlagensicht RPC“ die alleinige Sicht auf den RPC-Container eingestellt.

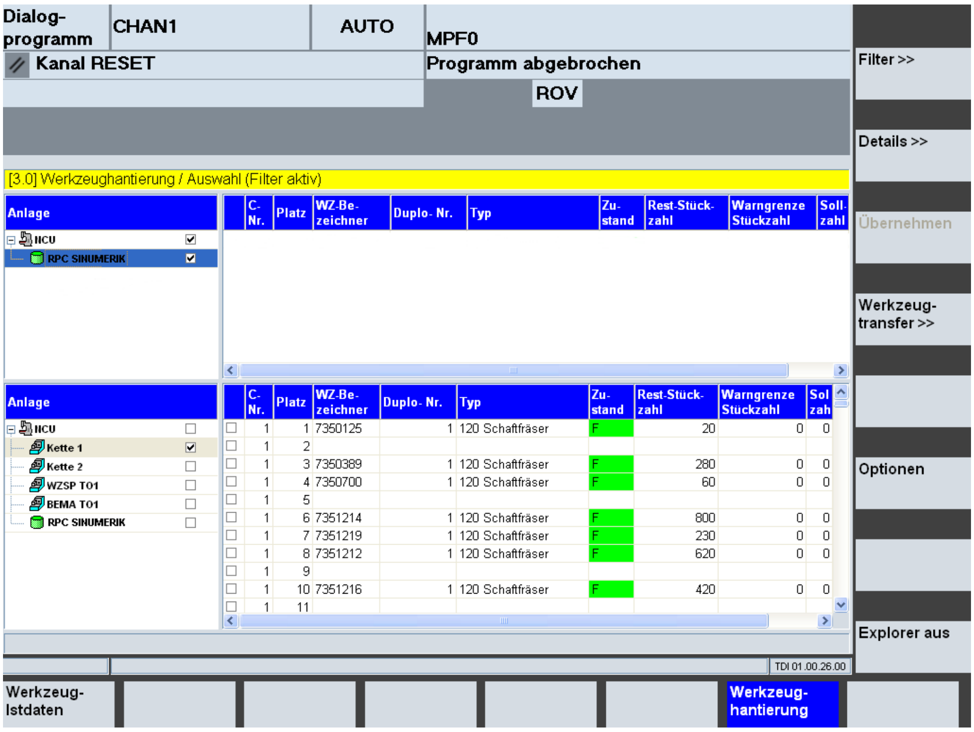


Bild 3-11 Werkzeughantierung/Filter

Die Anforderung der Werkzeugdaten vom FLR erfolgt dann in der Maske Filter.

Mit dem Softkey „Filter>>“ wird in die Maske zur Definition von Selektionsfiltern gewechselt.

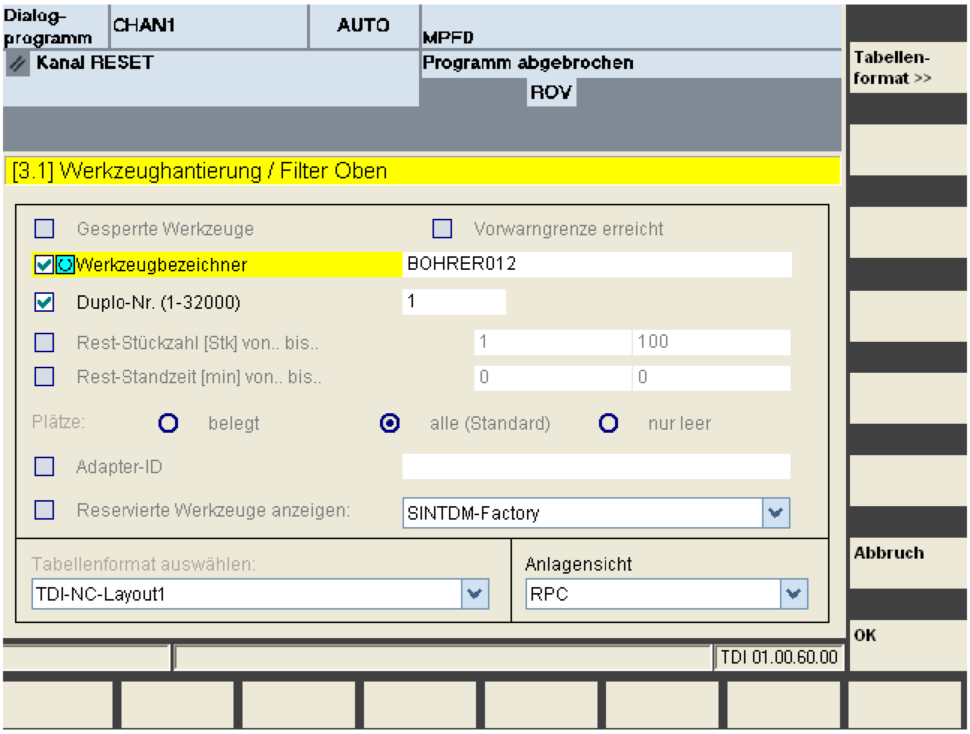


Bild 3-12 Definition von Selektionsfiltern

Hier müssen die Auswahl für Werkzeugbezeichner und Duplo-Nr. selektiert und die Daten für das anzufordernde Werkzeug eingegeben werden.
Mit dem Softkey „OK“ werden die Daten vom Leitrechner angefordert und es wird in die Maske Werkzeughantierung gewechselt.

Nachdem der Leitrechner die angeforderten Werkzeugdaten geliefert hat, werden diese im RPC-Container angezeigt.
Das Werkzeug kann nun Beladen werden (siehe Funktionsbeschreibung MCIS-TDI Kapitel „TDI Toolhandling“).

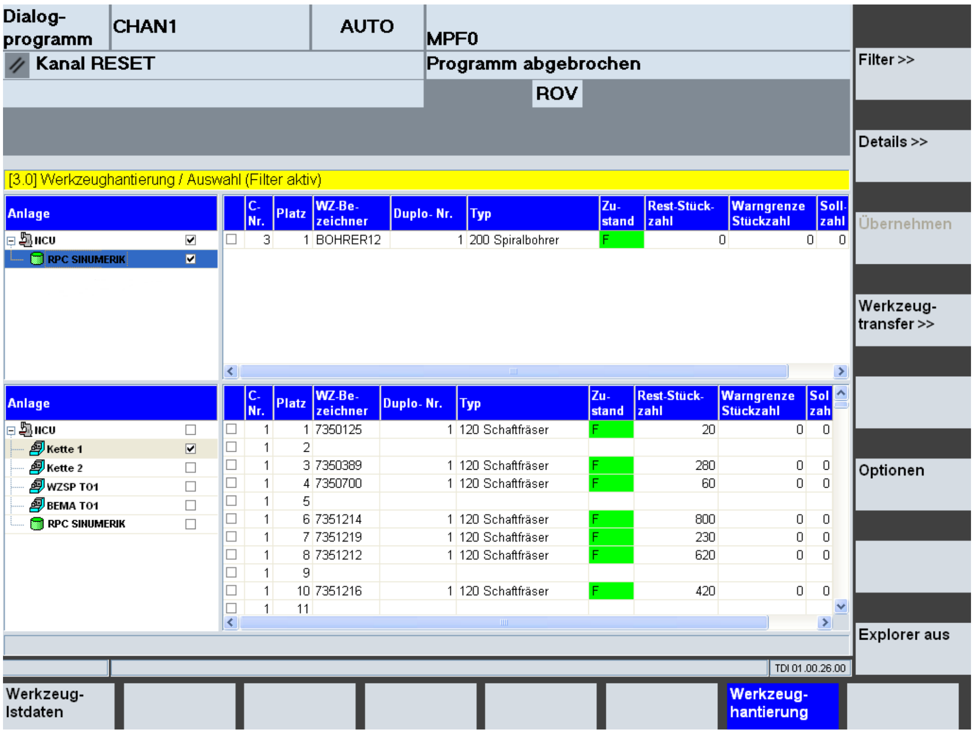


Bild 3-13 Werkzeughantierung/Filter: Anforderung abgeschlossen

Wurde der Beladevorgang aus dem RPC-Container erfolgreich abgeschlossen, werden die Werkzeugdaten mit Magazin und Platznummer an den FLR gemeldet.

Hinweis

Das Fileformat für die Werkzeugdaten entspricht dem Format der Datensicherung der NC840D (Lochstreifen-/ASCII-Format lt. /BA/ ; z.B. wie in _N_TOx_TOA oder _N_TOx_INI). Die vollständige Beschreibung zu Dateninhalt und Anordnung steht in /NFL/ Kap. 4 Werkzeugdaten.

Werkzeugdaten an FLR (Entladen von Werkzeugen)

Beim Entladen von Werkzeugen über den RPC-Container werden die Werkzeugdaten an den FLR gemeldet (Werkzeughantierung siehe Funktionsbeschreibung MCIS TDI Titel „TDI Toolhandling“).

Hinweis

Das Fileformat für die Werkzeugdaten entspricht dem Format der Datensicherung der NC840D (Lochstreifen-/ASCII-Format lt. /BA/ ; z.B. wie in _N_TOx_TOA oder _N_TOx_INI). Die vollständige Beschreibung zu Dateninhalt und Anordnung steht in /NFL/ Kap. 4 Werkzeugdaten.

Sind mehrere Werkzeuge über den RPC-Container entladen worden, so werden die Werkzeugdaten für jedes Werkzeug separat zum FLR gemeldet.

Hinweis

Nachdem die Werkzeugdaten erfolgreich zum FLR gemeldet wurden, wird das Werkzeug im RPC-Container gelöscht.

■

Für Notizen

4

4 Nahtstelle zwischen RPC SINUMERIK und TPS-PLC

4.1 Beschreibung

Zur Kommunikation zwischen RPC SINUMERIK und TPS-PLC wird ein Nahtstellen-DB benötigt. Hierfür wird ein Siemens-Standard-DB (DB12) reserviert. Der DB wird vom Anwender eingerichtet. Die Datenelemente der DB-Nahtstelle werden in Blöcken zusammengefasst, die jeweils einen Aspekt der Nahtstelle abdecken (z.B. Globaldaten, Transportauftrag ,Haltestellendaten). Die Darstellung der einzelnen Blöcke erfolgt in Tabellenform. Alle Blöcke werden aneinanderfolgend im Nahtstellen-DB abgelegt.
Die binären Datenelemente vom Typ "int(WORD)" bzw. "Long(DWORD)" werden im S7-Format (Little-Endian) im DB abgelegt. Bei Zugriffen von der PLC erfolgt dort eine entsprechende Wandlung in das Intel-Format (Big-Endian). Datenelemente - die einen Bezeichner darstellen - werden als Byte-Felder mit ASCII-Zeichen realisiert.
Die Beschreibung der Nahtstelle erfolgt in Tabellenform, in der Spalte „Zugriff von“ steht, wer dieses Feld beschreibt. Hierbei werden folgende Kürzel verwendet:

- RPC SINUMERIK Rechnerkopplungssoftware (indirekt vom FLR)
- PLC Anwender PLC-Programm

Um den internen Kommunikationsaufwand gering zu halten, wird jede Änderung in der Nahtstelle von RPC SINUMERIK mittels der Anforderung von PLC (Teil der Nahtstelle) mitgeteilt. RPC SINUMERIK richtet einen Hotlink auf dieses Datenelement ein.

4.2 Globaldaten

Tabelle 4-1 Globaldatenliste

Datenelement	**Kurzname**	**Daten-typ**	**Zugriff von**	**Offset**
Anforderung von PLC	ANF_PLC	Byte	PLC/RPC SINUMERIK	0
Änderungs-Trigger	Trigger	Byte	PLC	1
Anforderung von RPC SINUMERIK	ANF_RPC	Byte	RPC SINUMERIK	2
Anzahl der Transportaufträge	ANZ_TO	Byte	PLC	3
Anzahl der Haltestellen	ANZ_HALT	Byte	PLC	4
Maschinenzustand	Machine Status	Byte	PLC	5
NC-Betriebsart	Machine Mode	Byte	PLC	6
Maschinen-Modus	MODE	Byte	PLC	7
Reserve 1	Reserve 1	Word	PLC	8-9
Reserve 2	Reserve 2	Word	PLC	10-11

Anforderung von PLC

Tabelle 4-2 Zustandsliste von "Anforderung von PLC"

Bit-Nr.	Funktion	Zugriff von
0	Transportauftragsdaten geändert	PLC - 1/RPC SINUMERIK - 0
1	Haltestellendaten geändert	PLC - 1/ RPC SINUMERIK - 0
2	Zustandsänderung	PLC - 1/ RPC SINUMERIK - 0
3	Einzelplatz melden	PLC - 1/ RPC SINUMERIK - 0

Die PLC zeigt mit diesem Byte Änderungen in der Nahtstelle an. Die PLC muss, nachdem sie das Anforderungsbyte gesetzt hat, im Triggerbyte das nächste Bit setzen (siehe unten).
Die PLC darf in das Anforderungsbyte erst dann wieder schreiben, wenn es von RPC SINUMERIK nach der Verarbeitung auf 0 gesetzt wurde.

Transportauftragsdaten geändert

Transportauftragsdaten geändert wird von der PLC gesetzt, wenn bei einem der Transportaufträge der Status von der PLC geändert wurde.

Haltestellendaten geändert

Haltestellendaten geändert wird von der PLC gesetzt, wenn bei einer der Haltestellen Daten von der PLC geändert wurden.

Zustandsänderung

Zustandsänderung wird von der PLC bei jeder Zustandsänderung (Maschinen-Modus, Maschinenzustand, NC-Betriebsart) gesetzt über die der FLR informiert werden soll. RPC SINUMERIK muss daraufhin **R_TPS_H ()** an den FLR senden.

Einzelplatz melden

Die Haltestelle ist von der PLC in Reserve1 einzutragen. Die Haltestelle wird mit R_TPS_H an den FLR gemeldet.

Änderungstrigger

Auf Änderungen dieses Bytes reagiert die RPC SINUMERIK unmittelbar.
Die PLC setzt in diesem Byte jeweils ein Bit, wenn Änderungen von der PLC anstehen. Für jeden neuen Anstoß muss die PLC das nächste Bit setzen und das bisherige zurücksetzen, nach Bit 7 wird wieder mit Bit 0 begonnen.

Anforderungen von RPC SINUMERIK

Tabelle 4-3 Zustandsliste "Anforderung von RPC SINUMERIK"

Bit-Nr.	Funktion	Zugriff von
0	Synchronisationskennung	RPC SINUMERIK
1	Komponente abschalten	RPC SINUMERIK
2	Komponente einschalten	RPC SINUMERIK
3	Schreibzugriff auf Haltestellendaten anfordern	RPC SINUMERIK
4	Projektspezifische Sonderfunktion 1	RPC SINUMERIK
5	Projektspezifische Sonderfunktion 2	RPC SINUMERIK

Synchronisationskennung

Synchronisationskennung, wird vom Leitrechner gesetzt und zurückgenommen. (Siehe /NFL/ Kap. 5.14 C_SYNCH_M ()) Der Zustand des Transportsystems muss für die Dauer der Synchronisation unverändert bleiben. Die PLC darf keine neue Palettenbewegungen durchführen.

Komponenten abschalten

Komponenten abschalten, wird vom Leitrechner gesetzt (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung C_MODE_M ()). Es dient als Anforderung an die PLC Komponenten (Antriebe) abzuschalten. Siehe auch Maschinen Modus: **Komponenten abgeschaltet** .

Komponenten einschalten

Komponenten einschalten, wird vom Leitrechner gesetzt (Siehe /NFL/ Kap. 5.15 Mode-Umschaltung C_MODE_M ()). Es dient als Anforderung an die PLC Komponenten (Antriebe) wieder einzuschalten.

Schreibzugriff auf Haltestellendaten erlaubt

Schreibzugriff auf Haltestellendaten erlaubt wird von der PLC als Reaktion auf die Anforderung **Schreibzugriff auf Haltestellendaten anfordern** gesetzt. Durch diese Koordination wird verhindert, dass RPC SINUMERIK auf Grund einer Palettenbewegung falsche Haltestellen beschreibt.

Anzahl der Transportaufträge

Anzahl der Transportaufträge wird bei der Inbetriebnahme statisch hinterlegt. Sie gibt die max. Anzahl der Transportaufträge, die an die PLC vom FLR übergeben werden, vor. Sie entspricht der Anzahl der Transportauftragsdaten-Blöcke in der Nahtstelle.

Anzahl der Haltestellen

Anzahl der Haltestellen des Transportsystems wird bei der Inbetriebnahme statisch hinterlegt. Sie entspricht der Anzahl der Haltestellendaten-Blöcke in der Nahtstelle.

Maschinenzustand

Tabelle 4-4 Zustandsliste "Maschinenzustand"

Bit-Nr.	Funktion	Zugriff von
0	Maschine ist tätig	PLC
1	Maschine ist gestört	PLC
2	Neuanlauf der Maschine	PLC

NC-Betriebsart

Tabelle 4-5 Zustandsliste "NC-Betriebsart"

Bit-Nr.	Funktion	Zugriff von
0	Automatik	PLC
1	MDA	PLC
2	JOG	PLC
3	TEACH IN	PLC

Der Maschinenzustand und die NC-Betriebsart wird an den FLR mit R_TPS_H gemeldet. Sie werden jedoch nicht im RK-Server ausgewertet.

Maschinen-Modus

Tabelle 4-6 Zustandsliste "Maschinen-Modus"

Bit-Nr.	Funktion	Zugriff von
0	FLR-Mode	PLC
1	Komponenten abgeschaltet	PLC
4	Offline FLR1	RPC SINUMERIK
6	Wird bei R_TPS_H als Machine Mode 1000 gemeldet	PLC
7	Wird bei R_TPS_H als Machine Mode 2000 gemeldet	PLC

FLR-Mode

FLR-Mode wird von der PLC bzw. über einen Bediendialog gesetzt und zurückgenommen.

Komponenten abgeschaltet

Komponenten abgeschaltet wird von der PLC gesetzt, wenn dieser Zustand erreicht ist. Die Anforderung dazu erfolgt durch Bit 1 in den Anforderungs-kennungen (siehe unten).

Reserve1, Reserve2

Diese Variablen können vom Maschinenhersteller mittels der PLC beliebig benutzt werden. Die Werte werden bei R_MACHINE_H () an den FLR gemeldet, wobei im Standard im FLR keine Verarbeitung für diese Werte stattfindet.

4.3 Transportauftrag

Tabelle 4-7 Daten des Transportauftrages

Datenelement	Kurzname	Datentyp	Zugriff von
Quellhaltestelle	SDockIdx	Word	RPC SINUMERIK
Zielhaltestelle	DDockIdx	Word	RPC SINUMERIK/PLC
Werkstückträger	WPC	Byte[6]	RPC SINUMERIK
Werkstückträgertyp	WPCTyp	Byte	RPC SINUMERIK
Zur Pufferung	BufferFlag	Byte	RPC SINUMERIK
Priorität	Priority	Byte	RPC SINUMERIK
Verkettungsnummer	ChainNum	Byte	RPC SINUMERIK
Wagen	Vehicle	Byte	RPC SINUMERIK/PLC
Transportstatus	TPOStatus	Byte	RPC SINUMERIK/PLC

Quell- und Zielhaltestelle

Bei Quell- und Zielhaltestelle handelt es sich hier nicht mehr um die Haltestelle, wie sie im FLR geführt wird, sondern um einen Index. Dieser Index entspricht der Position dieser Haltestelle in den Haltestellendaten des Transportsystems. Die PLC muss die Koordinaten zu dieser Haltestelle mit dem selben Index verwalten. Die Umsetzung wird von RPC SINUMERIK anhand einer Zuordnungsliste in der Ini-Datei durchgeführt.

Werkstückträger

Die Bezeichnung des Werkstückträgers kann von der PLC zu Plausibilitätsprüfungen benutzt werden.

Werkstückträgertyp

Der Werkstückträgertyp ist eine Zusatzinformation, sie kann die Art oder die Größe eines Werkstückträgers enthalten.

Zur Pufferung

Die Kennung **zur Pufferung** wird gesetzt, wenn ein Werkstückträger auf eine Maschine gefahren wird, er aber dort nicht bearbeitet werden soll (Hilfspufferplatz). Diese Information muss vom TPS an die Maschine übergeben werden.

Priorität

Priorität ist eine Zusatzinformation. Werden mehrere Aufträge übergeben, kann durch die Priorität die Reihenfolge ihrer Abarbeitung beinflusst werden.

Verkettungsnummer

Verkettungsnummer ist eine Zusatzinformation. Bei Transportwagen mit zwei Liegeplätzen und Maschinen mit nur einer Haltestelle können zwei Transportaufträge durch eine Verkettungsnummer logisch verknüpft werden.

Wagen

Wagen ist eine Zusatzinformation. Bei Transportsystemen mit mehreren Transportwagen kann vorgegeben werden, mit welchem Wagen der Transport durchgeführt werden soll.

Transportstatus

Tabelle 4-8 TPOStatus

Bit-Nr.	Funktion	Zugriff von
0	Neuer Transportauftrag	RPC SINUMERIK
1	Transportauftrag begonnen	PLC
2	WPC auf Wagen	PLC
3	Auftrag abgeschlossen	PLC
4	Fehler, Auftrag nicht ausführbar	PLC
5	Fehler, Ersatzziel angefahren (in DDockPos ist das neue Ziel von der PLC eingetragen)	PLC

Mit Erteilung eines neuen Transportauftrages wird von RPC SINUMERIK im TPOStatus das Bit 0 gesetzt und Bit 1 bis Bit 7 gelöscht.
Bit 1 bis Bit 5 werden je nach Status der Abarbeitung von der PLC Bitweise gesetzt.

4.4 Haltestellendaten des Transportsystems

Tabelle 4-9 Haltestellendaten

Datenelement	Kurzname	Datentyp	Zugriff von
1. Haltestellenstatus	DockPosStatus	1 Byte	PLC
1. Werkstückträgerstatus	WPCStatus	1 Byte	PLC
1. Werkstückträger	WPC	6 Byte	PLC
...			

Die Anzahl der Haltestellen ist konfigurierbar, daraus ergibt sich die benötigte Länge im DB.

Haltestellenstatus

Tabelle 4-10 Haltestellenstatus

Bit-Nr.	Funktion	Zugriff von
0	Gestört	PLC
1	Gesperrt	PLC
2	Belegt	PLC

Das Bitfeld beschreibt den aktuellen Zustand der Haltestelle. Er wird von der PLC gesetzt. Ist kein Bit gesetzt, ist die Haltestelle freigegeben. Das Bit **Gestört** wird auf Grund von Peripherie-Signalen gesetzt bzw. zurückgenommen. Die Störungsursache wird über die Funktion Meldungen (Siehe /NFL/ Kap. 5.5 R_REPORT_H) dem Leitrechner mitgeteilt. Die PLC führt keine Palettentransporte zwischen Plätzen, die **Gestört** sind, aus. Ist die Haltestelle **Gesperrt**, darf sie nicht vom Transportsystem angefahren werden.
Belegt kann vor der PLC, vor allem für interne Zwecke, gesetzt werden. Der FLR (und auch RPC SINUMERIK) kann am Feld Werkstückträger erkennen, ob die Haltestelle belegt ist oder nicht.

Werkstückträgerstatus

Unbenutzt.

Werkstückträger

Bezeichner des Werkstückträgers, der sich aktuell auf der Haltestelle befindet (z.B. „WST01“). Diese Information wird von der PLC eingetragen. Ist kein Werkstückträger auf der Haltestelle, muss das Feld mit binär 0 gefüllt werden.

4.5 Transportauftrag an TPS

4.5.1 Funktionsablauf

Nahtstelleneintrag

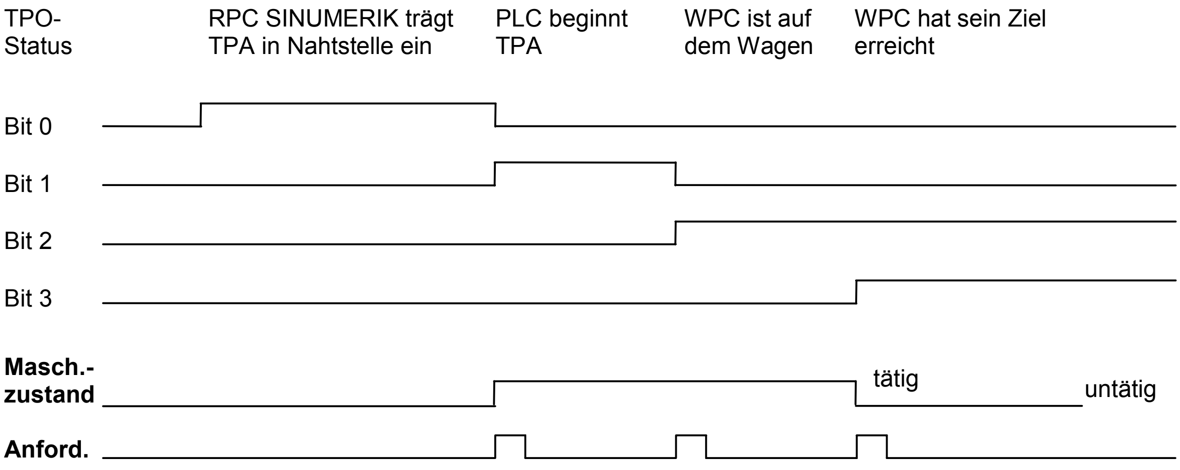
RPC SINUMERIK trägt die vom FLR erhaltenen Daten in den ersten freien Datensatz der Nahtstelle für Transportaufträge ein. Bei Ziel-und Quellhaltestelle muss eine Umsetzung auf einen Index erfolgen, da der PLC die Haltestellen-bezeichnung des FLR nicht bekannt ist.

PLC-Aktionen

Wenn die Transport-PLC den Auftrag beginnt, muss sie TPOStatus = **Auftrag begonnen** setzen. Damit RPC SINUMERIK diesen Zustand ausliest und dem FLR mitteilt, muss in den Globaldaten im Byte Anforderungen von PLC Bit 0 gesetzt werden. RPC SINUMERIK setzt das Anforderungsbyte zurück und liest alle zu meldenden Daten aus der Nahtstelle aus und setzt einen R_TPS_H - Call an den FLR ab.
Wenn der Werkstückträger auf den Transportwagen übernommen ist, kann die PLC TPOStatus = **WPC auf Wagen** setzen und wieder Bit 0 im Byte Anforderungen von PLC setzen, dadurch wird wieder ein R_TPS_H - Call an den FLR abgesetzt. Dieser Zwischenzustand muss jedoch nicht zwingend gemeldet werden. Es könnte auch erst wieder am Ende des Transportes das Anforderungs-bit gesetzt werden. Wurde es gesetzt, wird wieder ein R_TPS_H - Call an den FLR abgesetzt. In DockPos[0] wird SDockPos übernommen. In DockPos[1] wird die Nummer des Transportwagens eingetragen.
Nachdem der Werkstückträger an seiner Zielhaltestelle abgeliefert wurde, muss die PLC TPOStatus = **Auftrag abgechlossen** setzen und wieder im Byte **Anforderungen von PLC** das Bit 0 setzen.

Kann die Transport-PLC einen Transportauftrag gar nicht ausführen, weil z.B. die Quellhaltestelle leer oder gesperrt ist, muss sie TPO-Status = **Fehler, Auftrag nicht ausführbar** setzen. Kann nur das angegebene Ziel nicht angefahren werden und der Werkstückträger wird deshalb an einer anderen Haltestelle abgelegt, so muss die Transport-PLC in DDockPos das neue Ziel eingetragen und TPO-Status = **Fehler, Ersatzziel angefahren** setzen.
Auch am Ende des Transportauftrages wird wieder ein R_TPS_H - Call abgesetzt mit MachineStatus = „**untätig**“, DockPos[0] = DDockPos etc.

Jedesmal, wenn TPOStatus geändert wird und diese Änderung mittels R_TPS_H () an den FLR gesendet werden soll, muss im Byte **Anforderungen von PLC** das Bit 0 gesetzt werden.



4.6 Manuelle Transporte durch den Bediener auf PLC-Ebene

RPC SINUMERIK kann Zustandsänderungen oder Werkstückträgerbewegungen dem FLR in jeder Betriebsart melden, wenn Verbindung zum FLR besteht. Dadurch können Werkstückträgerbewegungen, die im manuellen Modus vom Bediener auf PLC-Ebene durchgeführt werden, dem FLR gemeldet werden. Dazu muss das PLC-Programm die jeweilige Quell- und Zielhaltestellennummer in der Transportauftrags-Nahtstelle setzen und TPOStatus auf **Transportauftrag begonnen** bzw. **Auftrag abgeschlossen**. Wird danach im Byte Anforderungen von PLC das Bit 0 gesetzt, werden die Daten von RPC SINUMERIK ausgelesen und an den FLR gesendet. Die Bewegung des Werkstückträgers kann also im Anlagenabbild angezeigt werden.

■

Für Notizen

5

5 Konfigurationsdaten

5.1 Beschreibung

Die für die Rechnerkopplung notwendigen Konfigurationsdaten werden in die Registry eingetragen. Diese Daten können mit dem Programm SCCONFIG.EXE erstellt und geändert werden.

Hinweis

Aus Kompatibilitätsgründen wurden die Registry-Einträge nicht auf den neuen Namen RPC SINUMERIK umgestellt, sondern bleiben unter HKEY_LOCAL_MACHINE\Software\SIEMENS\SinCOM\... erhalten.

Die Konfigurationsdaten werden in maschinenbezogene Daten und rechnerbezogene Daten unterschieden. Sind an die Rechnerkopplung mehrere Rechner angebunden, so gibt es für jeden Rechner einen Datensatz mit rechnerbezogenen Daten (Name, IP-Adresse ...). Für jeden Rechner muss festgelegt werden, welcher RPC von diesem Rechner angenommen wird und welcher RPC an den Rechner gesendet wird.

5.2 Konfigurationsprogramm SCCONFIG

Maschinenbezogene Daten

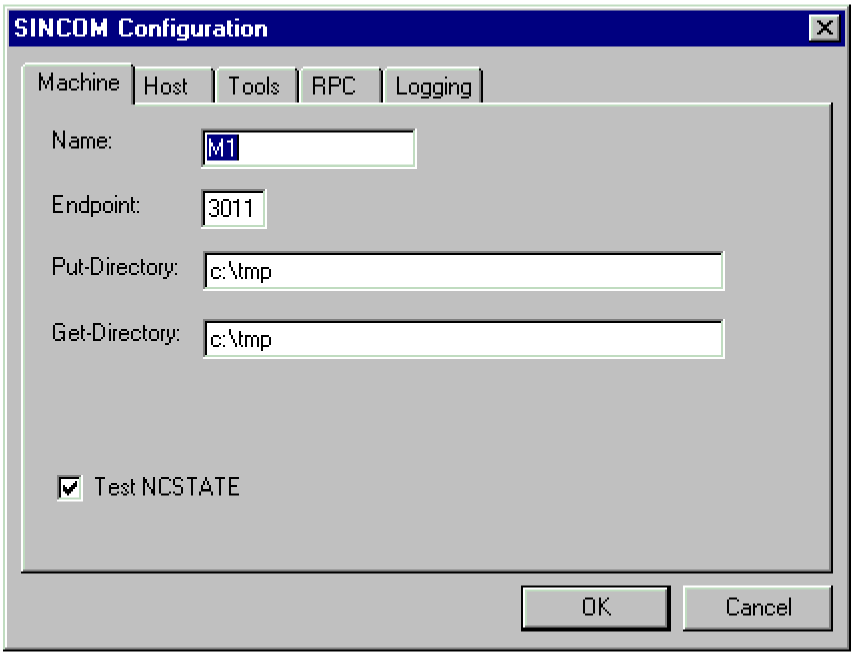


Bild 5-1 Konfigurationsprogramm: Register Machine

Name

Name ist der Maschinen-Name, der bei jedem RPC als zweiter Parameter eingetragen wird.

Endpoint

Die **Endpoint** Nummer wird bei RPC zusätzlich zur IP-Adresse gebraucht.

Put-Directory

Das **Put-Directory** dient als Temporärverzeichnis, in dem RPC SINUMERIK Dateien ablegt, bevor sie zum Leitrechner übertragen werden.)*

Get-Directory

Das **Get-Directory** dient als Temporärverzeichnis, wenn Dateien vom Leitrechner geholt werden. Wenn keine weitere Verarbeitung der Dateien durch RPC SINUMERIK stattfindet, bleiben sie in diesem Verzeichnis.)*

)* Es können sowohl Laufwerksbezeichnungen als auch die UNC-Notation verwendet werden.

Rechnerbezogene Daten

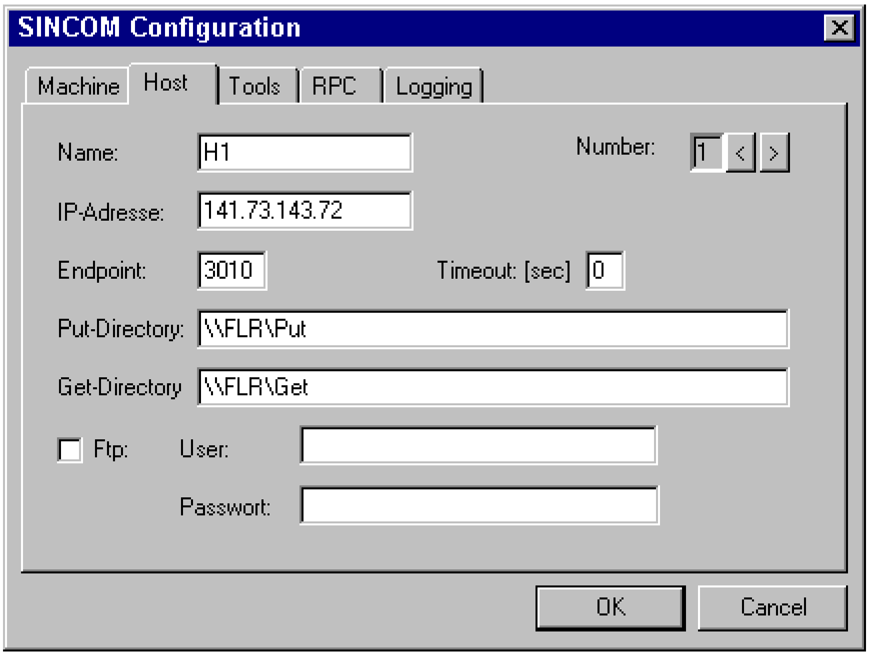


Bild 5-2 Konfigurationsprogramm: Register Host

Übersicht

RPC SINUMERIK bietet die Möglichkeit mit mehreren Leitrechnern zu kommunizieren. Für jeden Leitrechner müssen Sie diese Maske ausfüllen. Die Umschaltung zwischen den Leitrechnern erfolgt über die Pfeile oben rechts.

Host

Name ist der **Host-**Name, der bei jedem RPC Call als erster Parameter eingetragen wird.

IP-Adresse

Bei **IP-Adresse** ist die IP-Adresse des Leitrechners einzutragen.

Put-Directory

Das **Put-Directory** ist das Verzeichnis, in das RPC SINUMERIK Dateien auf den Leitrechner überträgt.

Get-Directory

Das **Get-Directory** ist das Verzeichnis, von dem RPC SINUMERIK Dateien auf dem Leitrechner holt.

Ftp

Handelt es sich beim Leitrechner um einen Windows Rechner (NT oder Win95), muss **Ftp** nicht verwendet werden, die Felder **User** und **Passwort** sind dann nicht relevant, da die Datenübertragung über SMB (Microsoft Notation) erfolgen kann. Bei anderen Leitrechnern z.B UNIX oder Linux Rechnern, muss die Dateiübertragung mittels FTP erfolgen, soweit nicht z.B. Samba auf den Leitrechnern installiert ist, das wiederum eine Microsoft Notataion erlaubt. Das Kästchen **Ftp** muss dann angewählt werden und bei **User** und **Passwort** sind gültige Einträge für die Anmeldung beim Leitrechner einzutragen. Der angegebene User muss Schreibrechte auf die bei **Put-Directory** und Leserechte auf die bei **Get-Directory** angegebenen Verzeichnisse haben.

Verzeichnisse

Die angegebenen Verzeichnisse auf dem Leitrechner und auch auf der PCU 50/70 **müssen existieren**, bzw. vor dem ersten Start von RPC SINUMERIK angelegt werden. Sie werden **nicht** von RPC SINUMERIK eingerichtet!

Werkzeugdaten lesen

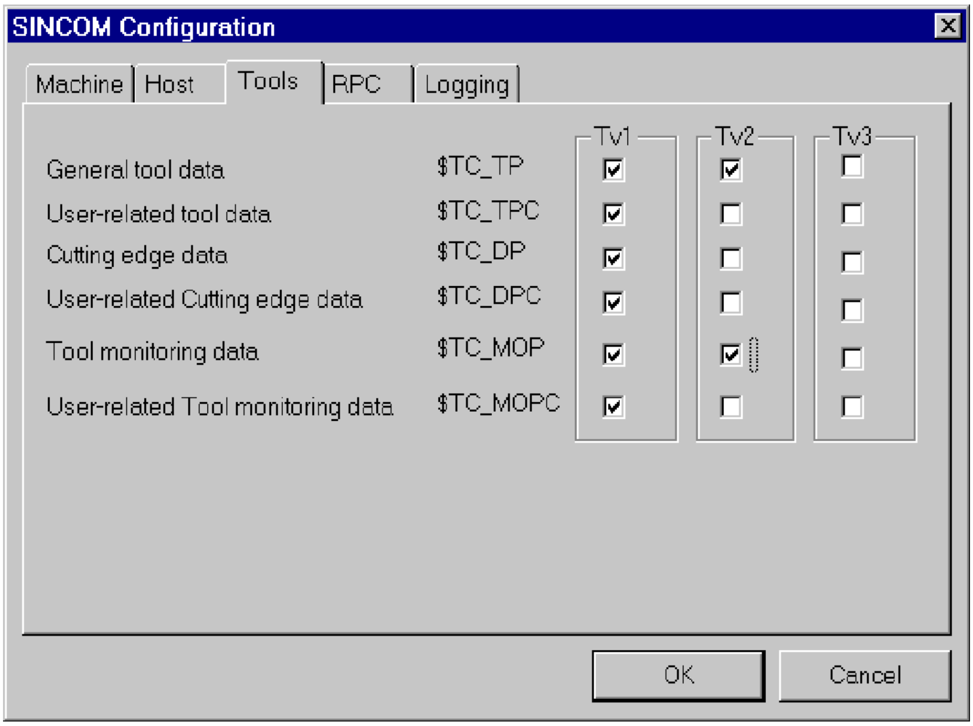


Bild 5-3 Konfigurationsprogramm: Register Tools

Das Lesen von Werkzeugdaten eines Werkzeuges aus der NC erfordert pro Datenbereich einen Leseaufruf.
Wenn Werkzeugdaten an den Leitrechner gemeldet werden, ist es nicht in jedem Fall notwendig alle WZ-Daten an den Leitrechner zu melden. Aus diesem Grund können drei Werkzeugstrukturen definiert werden (*WZ1-WZ3*). Die erste sollte in aller Regel alle **vorhandenen** Datenbereiche enthalten (sind keine anwender-bezogenen Datenbereiche vorhanden, sollten sie abgewählt werden). Die beiden weiteren sollten je nach Anforderung mit einem reduzierten Umfang definiert werden.

RPC-Funktionen

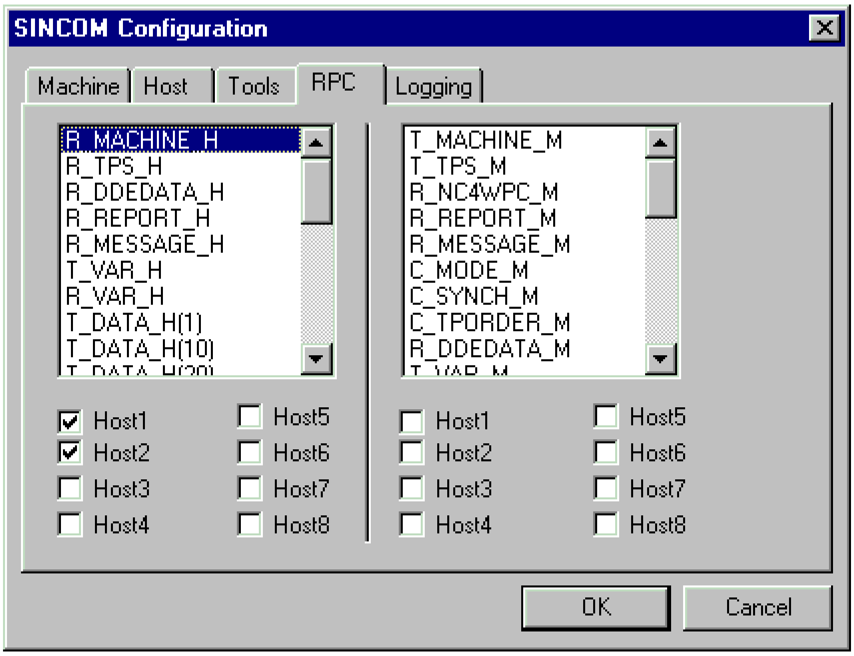


Bild 5-4 Konfigurationsprogramm: Register RPC

Für jede RPC-Funktion muss festgelegt werden, an welchen Leitrechner (Host) sie gesendet wird (Maske RPC, linke Seite) und welche RPC-Funktion von welchem Leitrechner entgegengenommen wird (Maske RPC, rechte Seite).

Logging

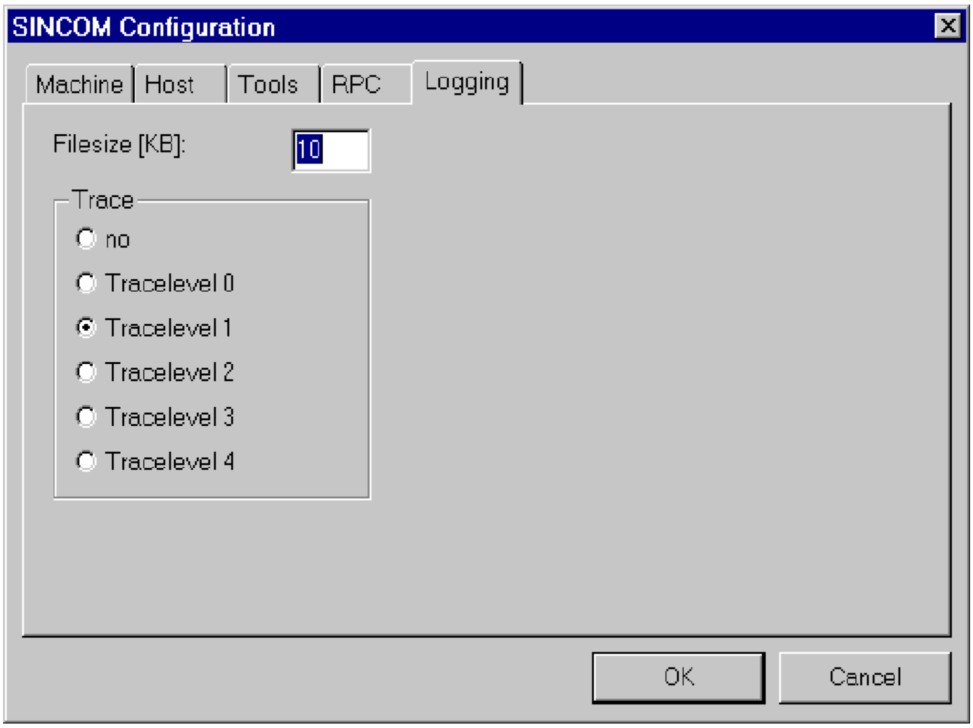


Bild 5-5 Konfigurationsprogramm: Register Logging

Der Dialog für Logging erlaubt die Festlegung der Größe der Logdatei und die Wahl eines Tracelevels.

5.3 Konfigurationsdaten Beispiel

Die ASCII-Datei mit Konfigurationsdaten könnte wie folgt aussehen:

```
[HKEY_LOCAL_MACHINE\Software\SIEMENS\SinCOM\Host1]
"Name"="FLR1"
"IpAdr"="195.212.26.110"
"Endpoint"="2010"
"FTPUser"=""
"FTPPassword"=""
"HostDirGet"="h:\\"
"HostDirPut"="h:\\"
"Ftp"=dword:00000000

[HKEY_LOCAL_MACHINE \Software\SIEMENS\SinCOM\RPC_H]
"R_MACHINE_H"=dword:0000ffff
"R_TPS_H"=dword:0000ffff
"R_DDEDATA_H"=dword:0000ffff
"R_REPORT_H"=dword:0000ffff
"T_VAR_H"=dword:00000000
"R_VAR_H"=dword:00000000
"T_DATA_H(1)"=dword:00000001
"T_DATA_H(10)"=dword:00000001
"T_DATA_H(20)"=dword:00000001
"T_DATA_H(21)"=dword:00000001
"T_DATA_H(22)"=dword:00000001
"T_DATA_H(23)"=dword:00000001
"T_DATA_H(26)"=dword:00000001
"T_DATA_H(27)"=dword:00000001
"T_DATA_H(28)"=dword:00000001
"T_DATA_H(90)"=dword:00000001
"R_DATA_H(1)"=dword:000000ff
"R_DATA_H(10)"=dword:000000ff
"R_DATA_H(20)"=dword:000000ff
"R_DATA_H(21)"=dword:00000001
"R_DATA_H(22)"=dword:00000001
"R_DATA_H(23)"=dword:00000001
"R_DATA_H(50)"=dword:00000001
"R_DATA_H(90)"=dword:00000001

[HKEY_LOCAL_MACHINE \Software\SIEMENS\SinCOM\RPC_M]
"T_MACHINE_M"=dword:0000ffff
"T_TPS_M"=dword:0000ffff
"T_DATA_M"=dword:00000001
"T_REPORT_M"=dword:00000001
"R_DATA_M"=dword:00000001
"R_NC4WPC_M"=dword:00000000
"R_DDEDATA_M"=dword:0000ffff
"R_REPORT_M"=dword:0000ffff
"C_DELETE_M"=dword:00000001
"C_TPORDER_M"=dword:00000001
"R_DATA_M(1)"=dword:00000001
"R_DATA_M(10)"=dword:00000001
"R_DATA_M(26)"=dword:00000001
```

```
"R_DATA_M(27)"=dword:00000001
"R_DATA_M(28)"=dword:00000001
"R_DATA_M(90)"=dword:00000001
"C_DELETE_M(1)"=dword:00000001
"C_ORDER_M(1)"=dword:00000001

[HKEY_LOCAL_MACHINE\Software\SIEMENS\SinCOM\Settings]
"Machine"="FMS-TPS________"
"EndpointMach"="3011"
"MMCDirGet"="C:\\TMP\\MMC"
"MMCDirPut"="C:\\TMP\\MMC"
"TraceFilesize"=dword:00000005
"TraceLevel"=dword:00000002
"ToolData1"=dword:0000003f
"ToolData2"=dword:00000015
"ToolData3"=dword:00000001
```

■

I

I Index

I.1 Stichwortindex

■

Für Notizen

A

A Anhang

A.1 Interface Definition Language (IDL)

Hinweise zur Handhabung

Auf der Installations-CD der RPCTEST Installation sind die Dateien

- SCHOST.IDL
- SCMACH.IDL
- SCHOST.ACF
- SCMACH.ACF

enthalten.

Die IDL Dateien beschreiben die Funktionsaufrufe mit ihren Parametern. Mit den ACF Dateien wird für den IDL-Compiler festgelegt, ob man internal- oder external binding will. Wenn der FLR mit mehreren Maschinen kommunizieren soll, muss external binding benutzt werden.
Die mitgelieferten ACF Dateien enthalten für beide Arten des Bindings die entsprechende Anweisung, eine davon ist jedoch auskommentiert. Sorgen Sie dafür, dass die gewünschte Form aktiv und die unerwünschte Kommentar ist.
Der IDL-Compiler generiert aus diesen Dateien die Client- und Server-Stubs sowie Headerdateien.
Wenn auf dem FLR Microsoft WINDOWS (>= WINDOWS 95) zum Einsatz kommt, reicht das VC++ Entwicklungssystem aus. Es enthält einen IDL-Compiler und die sonstigen für RPC notwendigen Dateien.

Aufruf des Microsoft IDL Compilers:

MIDL SCHOST.IDL /osf	für die RPCs an den FLR (z.B. R_MACHINE_H)
MIDL SCHMACH.IDL /osf	für die RPCs an SINCOM (z.B. T_MACHINE_M)

Der Paramter /osf muss angegeben werden, um DCE-kompatibel zu sein.

Bei anderen Betriebsystemen wird ein DCE-RPC Entwicklungssystem benötigt.

! **Wichtig**

Es muss ein DCE RPC Entwicklungssystem sein und nicht SUN RPC!

Die RPCTEST Installation enthält im Verzeichnis EXAMPLE die Sourcen eines VC++ 6.0 Musterprogrammes als Vorlage für eine RPC SINUMERIK Leitrechnerapplikation, sowie eine kurze Beschreibung der wichtigsten Programmteile.

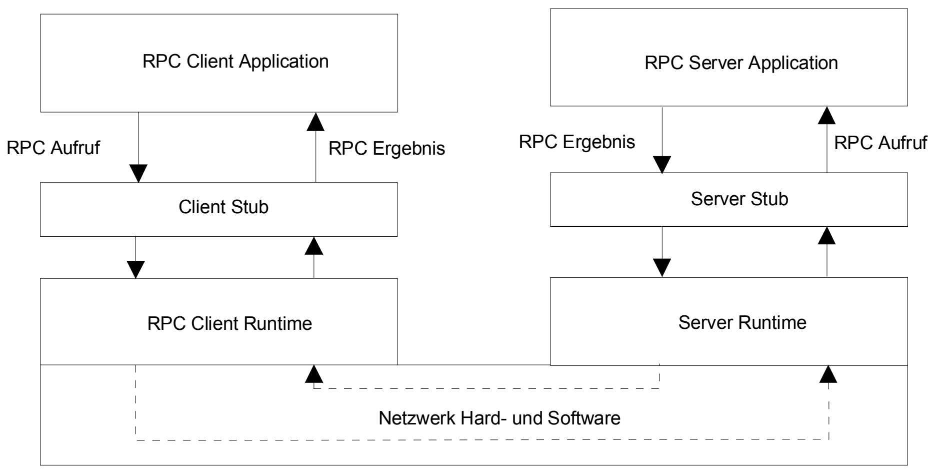


Bild A-1 Ablauf eines RPC Aufrufs

A.1.1 Funktionen zum Fertigungsleitrechner (FLR) - SCHOST.IDL

```
 [ uuid(d3d7d860-c15a-11d0-a0cb-00a0244ce687),
  version(1.0),
  pointer_default(unique)
]
interface SINCOMHOST
{

const long maxWPCLen = 6;
const long maxMPos = 3;

long R_MACHINE_H([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long MachineMode,
     [in] long MachineStatus,
     [in, string] unsigned char* pszNCProgramm,
     [in] long ClampCubeSide,
     [in] long DockPos[maxMPos],
     [in] long DockPosStatus[maxMPos],
     [in] unsigned char pszWPC[maxMPos][maxWPCLen],
     [in] long WPCStatus[maxMPos],
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );

const long maxTPos = 2;

long R_TPS_H([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long MachineMode,
     [in] long MachineStatus,
     [in] long TpOStatus,
     [in] long DockPos[maxTPos],
     [in] long DockPosStatus[maxTPos],
     [in] unsigned char pszWPC[maxTPos][maxWPCLen],
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );

const long maxAlarms = 10;

long R_REPORT_H([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long Typ,
     [in] long Number[maxAlarms],
     [in] long Time[maxAlarms],
     [in] char Flag[maxAlarms],
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );
```

```
long R_MESSAGE_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in, string] unsigned char* pszMessage,
    [in] long ResInt1,
    [in] long ResInt2,
    [in, string] unsigned char* pszResByte );

long T_DATA_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long SFkt,
    [in, string] unsigned char* pszName1,
    [in, string] unsigned char* pszName2 );

long R_DATA_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long SFkt,
    [in, string] unsigned char* pszName1,
    [in, string] unsigned char* pszName2,
    [in] long Date,
    [in] long LastFile );

long T_VAR_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long VarMode,
    [in, string] unsigned char* pszVarSet,
    [in, string] unsigned char* pszVarDescr );

long R_VAR_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long VarMode,
    [in, string] unsigned char* pszVarSet,
    [in, string] unsigned char* pszVarDescr,
    [in, string] unsigned char* pszVarData );

long R_DDEDATA_H([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in, string] unsigned char* pszData );

void Shutdown_H(void);

}
```

A.1.2 Funktionen zur SINUMERIK - SCMACH.IDL

```
[ uuid(d6542300-c15a-11d0-a0cb-00a0244ce687),
  version(1.0),
  pointer_default(unique)
]
interface SINCOMMACHINE
{

long T_MACHINE_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum);

long T_TPS_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum);

long T_DATA_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long SFkt,
     [in, string] unsigned char* pszName1,
     [in, string] unsigned char* pszName2 );

long T_VAR_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long VarMode,
     [in, string] unsigned char* pszVarSet,
     [in, string] unsigned char* pszVarDescr );

long R_NC4WPC_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in, string] unsigned char* pszWPC,
     [in, string] unsigned char* pszNCProg,
     [in] long Date,
     [in] long NCPLength,
     [in] long ClampCubeSide,
     [in] long TpFlag,
     [in] long NCExtern,
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );
```

```
long R_REPORT_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long Typ,
     [in] long Number,
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );

long R_MESSAGE_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in, string] unsigned char* pszMessage,
     [in] long ResInt1,
     [in] long ResInt2,
     [in, string] unsigned char* pszResByte );

long R_DATA_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long SFkt,
     [in, string] unsigned char* pszName1,
     [in, string] unsigned char* pszName2,
     [in] long Date,
     [in] long LastFile );

long R_VAR_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long VarMode,
     [in, string] unsigned char* pszVarSet,
     [in, string] unsigned char* pszVarDescr,
     [in, string] unsigned char* pszVarData );

long R_DDEDATA_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in, string] unsigned char* pszApplication,
     [in, string] unsigned char* pszTopic,
     [in, string] unsigned char* pszItem,
     [in, string] unsigned char* pszData );

long C_DELETE_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long SFkt,
     [in, string] unsigned char* pszName1,
     [in, string] unsigned char* pszName2     );

long C_MODE_M([in, string] unsigned char* pszHost,
     [in, string] unsigned char* pszMachine,
     [in] long OrderNum,
     [in] long Mode   );
```

```
long C_SYNCH_M([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long SynchFlag);

long C_TPORDER_M([in, string] unsigned char* pszHost,
    [in, string] unsigned char* pszMachine,
    [in] long OrderNum,
    [in] long SDockPos,
    [in] long DDockPos,
    [in, string] unsigned char* pszWPC,
    [in] long WPCTyp,
    [in] long BufferFlag,
    [in] long Priority,
    [in] long ChainNum,
    [in] long Vehicle,
    [in] long ResInt1,
    [in] long ResInt2,
    [in, string] unsigned char* pszResByte );

void Shutdown_M(void);

}
```

A.2 (OEM-) Schnittstelle HMI <=> NCK/PLC

Hinweis

Das folgende Unterkapitel ist unverändert der Dokumentation für (OEM-) Schnittstelle HMI <=> NCK/PLC entnommen.

8

(OEM-) Schnittstelle HMI <=> NCK/PLC

Übersicht

Dem Entwickler stehen drei unterschiedliche Dienste zur Kommunikation zur Verfügung:

Variablendienst	Zugriff auf NC-, PLC- und Antriebsdaten	
	Über	**OPC-DataAccess**
	Oder	**DCTL-Control**
Domaindienst	Kopieren von Dateien zwischen HMI und NCK:	
	Filezugriff über Datenhaltung	
	Mit	**IMC-File**
	Und	**IADS**
	Oder	**FileViewer-Control**
Programm-Instanz-Dienst:		Funktionen in der NC starten
	Mit	**IMC-Command**

8.1 Allgemein

Die Kommunikation zwischen Anwendungen und NC/PLC erfolgt über die OPCbzw.
die Sinumerik-COM-Schnittstellen. Aus Gründen der Kompatibilität besteht vorerst weiterhin die Möglichkeit, direkt über den NCDDE-Server zuzugreifen. Bei Neuentwicklungen sollte aber unbedingt über die neuen COM-Schnittstellen zugegrifen
werden.

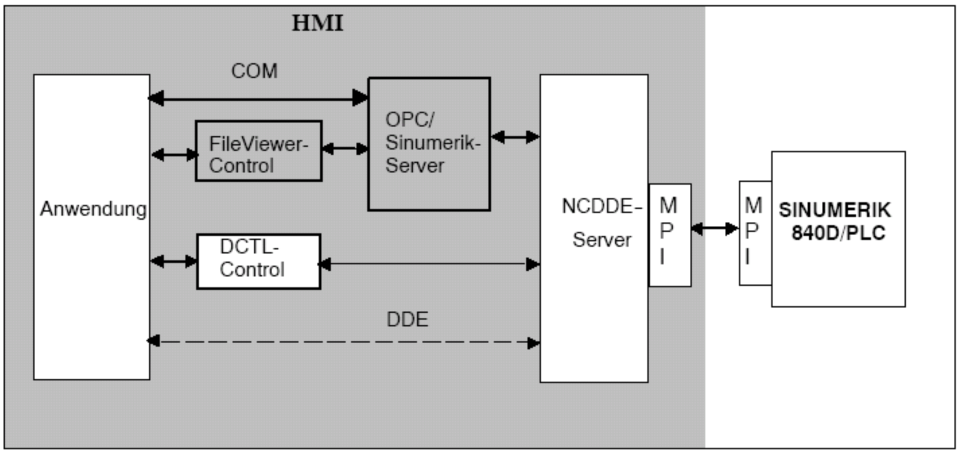


Bild 8-1 Übersicht

Der NCDDE-Server ist über Initialisierungs-Dateien konfigurierbar. Dies erlaubt es dem Anwender, den NCDDE-Server seiner Entwicklungsumgebung anzupassen. Kann an einer Steuerung getestet werden? Ist eine oder sind mehrere NCUs vorhanden, auf deren Daten zugegriffen werden soll ?

Hinweis

Durch den Einsatz der Windows Umgebung gibt es für die Kommunikation mit der
SINUMERIK 840D keine Zeitgarantie. Echtzeit-Aufgaben sind deshalb nicht im HMI lösbar. Sie sind im Bereich der NCU mit dem OEM-Paket NCK zu lösen.

8.2 DDE-Grundlagen

Übersicht

Das Betriebssystem WINDOWS stellt dem Anwendungs-Entwickler den DDE(Dynamic
Data Exchange)-Mechanismus als Kommunikationsmittel zwischen WINDOWS-Prozessen zur Verfügung.

DDE-Grundzüge

DDE kennzeichnet den dynamischen Datenaustausch unter WINDOWS mit folgenden
Eigenschaften:

- DDE ist Kommunikation zwischen WINDOWS-Applikationen
- DDE läuft zwischen zwei Prozessen nach dem Client-Server-Modell ab
- Ein Prozeß ist Client: Fordert von anderem Programm (dem Server) Daten an
- Ein Prozeß ist Server: Liefert dem anderen Programm (dem Client) die Daten
- Verbindung wird vom Client aufgebaut
- Ein Programm kann sowohl Client als auch Server sein
- Als Protokoll dient das WINDOWS-eigene Verfahren

DDE-Verbindungsaufbau

Um die Verbindung zu einem DDE-Server aufbauen zu können, muss dem Entwickler des Clients folgendes bekannt sein:

- Link-Server Name des DDE-Servers
- Link-Topic Thema
- Link-Item Datum, auf das zugegriffen werden soll
- Link-Mode Verbindungsart

DDE Link-Mode

- Request Das Datum wird einmal vom Client abgefragt
- Warmlink Der Server informiert den Client, wenn sich das Datum geändert hat. Der Client kann sich dann das Datum abholen
- Hotlink Wenn sich das Datum ändert, sendet der Server dem Client automatisch den aktuellen Wert des Datums
- Poke Der Client beauftragt den Server, ein Datum zu schreiben
- Execute Der Client beauftragt den Server ein Kommando auszuführen

8.3 NCDDE-Server Konfiguration

8.3.1 Die Initialisierungsdatei MMC.INI

Beschreibung

Die Initialisierung des NCDDE-Servers geschieht über die Sektion ”[GLOBAL]” in der Datei ”MMC.INI”. Diese Datei liegt im Verzeichnis ”\MMC2” des OEM-Systems. Hier werden Link-Server und Link-Topic bestimmt, zu denen der lokale NCDDEServer Verbindung aufnehmen soll. Die Begriffe Link-Server und Link-Topic werden im Kapitel 8.2 erläutert.
Je nach Installationsumfang kann der NCDDE-Server vier Grundkonfigurationen annehmen:

- Verbindungsaufbau zu einer NC
- Verbindungsaufbau zu einer oder mehreren NCs
  (M:N-Funktionalität siehe Kapitel 8.3.3)
- lokaler Betrieb auf einem PC
  Gibt dem Entwickler die Möglichkeit, lokal auf einem PC, ohne Verbindung zu einer NC, seine Applikation zu testen. Der NCDDE-Server gibt in diesem Fall Ersatzwerte aus, die mit dem Kommando ”NEW” (Kapitel 8.8) definiert und dem Kommando ”ANIMATE” (Kapitel 8.8) so verändert werden können, dass der Eindruck einer aktiven NC entsteht.
- lokaler Betrieb auf einem PC mit NC-Simulator.
  Gibt dem Entwickler die Möglichkeit, lokal auf einem PC, ohne Verbindung zu einer NC, seine Applikation zu testen. Mit dem NC-Simulator ist es möglich, ein NC-nahes Verhalten für den HMI nachzubilden.

NcddeServiceName

DDE-Link-Service Name des NCDDE-Server. Er ist auf ”ncdde” voreingestellt.

Hinweis

Alle Beispiele von Kapitel 8 beruhen auf dem ”NcddeServiceName= ncdde”.Hat dieser einen anderen Namen, müssen diese Beispiele entsprechend modifiziert werden, ansonsten funktionieren sie nicht.

Ncdde-MachineName

Hier wird der NCU-Name für die Standard-Applikationen eingetragen. Ist hier die Bezeichnung ”MachineSwitch” eingetragen, dann ist die Umschaltung zwischen einzelnen NCUs möglich (M:N Funktionalität Kapitel 8.3.3).

NcddeDefault-MachineName

Hiermit wird die Initialisierung für die M:N Funktionalität eingestellt, d.h. zu dieser NCU wird beim Hochfahren des HMI die Verbindung aufgebaut.

Ncdde-MachineNames

Hier werden die Namen der NCUs eingetragen, zu denen eine Verbindung aufgebaut werden kann. Die hier eingetragenen NCU-Namen **müssen** jeweils als Sektion mit diesem Namen in der Datei MMC.INI vorhanden sein.

NcddeStartupFile

NSK-Datei (Kapitel 8.3.2), die beim Start des NCDDE-Servers geladen werden soll.

NcddeMachineNamesAdd1

Kennzeichnung einer installierten NC-Simulation. Wenn keine Simulation installiert wurde, dann ist dieser Eintrag ohne Bedeutung.

Im folgenden Beispiel wird die Parametrierung der Datei MMC.INI durch eine Installation
auf einem PC ohne NC und ohne Simulation gezeigt.

Beispiel 8-1 Auszug aus der Datei MMC.INI

```
[GLOBAL]
; for using M:N function set NcddeMachineName=MachineSwitch
; for working without NC set NcddeMachineName=local
; for working with SIMNC set NcddeMachineName=SIM1
; for connecting to a NC set NcddeMachineName=NCU840D
NcddeMachineName=local
; for using M:N function set NcddeDefaultMachineName=net:NCU_1
; for working without NC set NcddeDefaultMachineName=local
; for working with SIMNC set NcddeDefaultMachineName=SIM1
; for connecting to a NC set NcddeDefaultMachineName=NCU840D
NcddeDefaultMachineName=local
; for using M:N function set NcddeMachineNames=net,NCU840D
; for working without NC set NcddeMachineNames=
; for working with SIMNC set NcddeMachineNames=SIM1
; for connecting to a NC set NcddeMachineNames=NCU840D
NcddeMachineNames=
; for using M:N function set NcddeStartupFile=ncdde5.nsk
; for working without NC set NcddeStartupFile=ncdde202.nsk
; for working with SIMNC set NcddeStartupFile=sim1dde5.nsk
; for connecting to a NC set NcddeStartupFile=ncdde5.nsk
NcddeStartupFile=ncdde202.nsk
```

Namensräume

Der NCDDE-Server trifft im LOCAL-Betrieb bei Variablen keine Unterscheidung in 'Namensräume'. Mit Namensraum ist eine Unterscheidung nach TOPIC gemeint. Wenn also zum TOPIC LOCAL eine Variable angelegt wurde und dieselbe Variable
zum TOPIC Sim0, dann werden diese Variablen vom NCDDE nicht unterschieden. Das führt z.B. dazu, dass die aktuelle Satzanzeige in Simulation nicht mehr funktioniert,
wenn man einmal nach MASCHINE in ein Bild mit aktueller Satzanzeige gewechselt
ist, da dort aus dem Programm heraus lokale Variablen angelegt werden
und diese die 'Simulations-Variablen' überdefinieren.

8.3.2 Kommando-Dateien des NCDDE-Servers

NSK-Dateien

Die Kommando-Dateien (mit der Endung .NSK) enthalten z.B. die Link-Items, auf die sich die NCDDE-Verbindungen beziehen. Diese Dateien können Kommandos enthalten, die in Kapitel 8.8 beschrieben sind.
In diesen Dateien werden die Daten-Link-Items, auf die zugegriffen werden kann, beschrieben. Es können aber auch weitere .nsk -- Files über CALL-Aufrufe mit einbezogen
werden. Auf diese Weise ist eine Strukturierung möglich. Beispiel 8-2
zeigt ein Link-Item (LastError) und die Strukturierung der globalen Variablen für HMI mit CALL-Anweisungen.

Hinweis

Der Anwender hat die Möglichkeit, eigene NSK-Dateien mit CALL-Anweisungen zu laden. Die NSK-Dateien können mit einer MAP--Funktion (Kapitel 8.6.3) erzeugt werden.

Beispiel 8-2 Die Datei NCDDE311.NSK im Verzeichnis \mmc2

```
REM NSK ROOT FOR 840D
============================================================
REM
REM WRITE-ACCESS FOR NC-BUSADDRESS
LINK(”/Nck/Nck/busAddress”,200,”7 31 0 0 E0# /NC 1 0 11”,10)
LINK(”/Nck/Nck/busState”,300,””,0);
REM
REM ACCESS TO CONNECTION ERROR STATE
LINK(”LastError”,1,””,0);
REM
REM IMPORT 840D BASIC NC VARIABLES
CALL(nc311.nsk)
REM
REM IMPORT 840D BASIC PLC VARIABLES
CALL(plc311.nsk)
REM
REM IMPORT ADDITIONAL LINK VARIABLES
CALL(add311.nsk)
REM
REM IMPORT COMIC STARTS
CALL(comic.nsk)
REM
```

8.3.3 Verbindung zu mehreren NCs

M:N–Funktionalität

Diese Funktionalität erlaubt die grundsätzlich die Verbindung mehrerer HMIs mit mehreren NCUs. Man kann z.B. von einem HMI auf Daten von zwei NCUs zugreifen. Die Datei “NETNAMES.INI“ wird bei dieser Grundkonfiguration interpretiert.

Verbindungsteil

Die Sektion “[conn MMC_1]“ legt fest, zu welchen Partnern der HMI Verbindung aufnehmen kann.

Netz–Parameter

In der Sektion “[param network]“ wird die Übertragungsrate festgelegt:

BTSS	1,5 MBit
MPI	187,5 KBit

Bus–Teilnehmer

In den Sektionen “[param NCU_n]“ werden die Busadressen der NC und der PLC sowie der Name der NCU festgelegt. Über diesen Namen muss die NCU vom HMI aus angesprochen werden. Pro NCU muss eine Beschreibung vorhanden sein.

Beispiel 8-3 Die Datei NETNAMES.INI

```
; owner TECHNISCHE Referenz auf Busbeschreibung
; Rechnerspezifisch
[own]
owner= MMC_1
; Description of possible connections
[conn MMC_1]
conn_1= NCU_1
conn_2= NCU_2
; Description of significant net-parameters
; btss =1,5MBit
; mpi =187,5 KBit
[param network]
bus= btss
; Busadressen für alle Busteilenehmer
[param MMC_1]
mmc_address= 1
[param NCU_1]
nck_address= 10
plc_address= 10
name=Standard_Machine
[param NCU_2]
nck_address= 11
plc_address= 11
name=Test_Maschine
```

Applikation

Damit sich OEM–Applikationen in einer M:N–Konfiguration korrekt verhalten, ist folgendes zu beachten:

- Für jegliche Kommunikation zur NC sollte als LinkTopic ausschließlich „machineswitch" verwendet werden. Dadurch ist sichergestellt, dass die Applikation stets mit der NC kommuniziert, die im Rahmen einer M:N– bzw. NC–Umschaltung angewählt wurde.
  Wird beim Einrichten einer Kommunikationsverbindung im LinkTopic dagegen der Name einer bestimmten NCU angegeben, z.B. NCU1, so wird diese im Rahmen einer M:N–Umschaltung nicht berücksichtigt, d.h. bei Umschalten der Bedientafel von NCU1 auf eine andere NC, bleibt diese Verbindung weiterhin bestehen. Solche „festen" bzw. statischen Verbindungen zu einer bestimmten NC dürfen von einer Applikation nur dann eingerichtet werden, wenn die Bedientafel, auf der die Applikation abläuft, als M:N–Server konfiguriert ist.
- Eine NC–Umschaltung stellt sich für die OEM–Applikation wie ein NC–Reset bzw. Kommunikationsausfall dar. Die Applikation sollte sich dementsprechend auch so verhalten wie nach NC–Reset bzw. Kommunikationsausfall. Einen NC–Reset erkennt man durch Einrichten eines Hotlinks auf die BTSS–Variable /bag/state/opmode. Im Fall eines NCK–Reset liefert der Hotlink den Wert „#".
- Um im Hintergrund ablaufende Kommunikationsvorgänge, z.B. Dateitransfers, durch eine M:N– bzw NC–Umschaltung nicht zu unterbrechen bzw. abzubrechen, sollte die NC–Umschaltung vor dem Start solcher Vorgänge gesperrt und nach deren Beendigung wieder freigegeben werden. Dazu stehen die Funktionen LockCurrentNCU/LockChanMenu bzw. UnlockCurrentNCU/UnlockChanMenu (siehe Kapitel 6.4). zur Verfügung. Während LockCurrentNCU nur die Umschaltung zu einer anderen NC sperrt, Kanalumschaltungen auf der aktuell angewählten NC aber weiterhin zuläßt, verhindert LockChanMenu auch die Kanalumschaltung.
- Falls bestimmte Dienste bzw. Variablen nur auf einer bestimmten NCU zur Verfügung stehen, ist vor dem Zugriff auf diese Dienste bzw. Variablen von der Applikation zu prüfen, ob aktuell eine Verbindung zu der betreffeden NCU besteht. Die NCU zu der aktuell eine Verbindung besteht, kann durch Auslesen der Variablen „machineswitch" (KinkItem) ermittelt werden.

8.4 Aufbau einer DDE–Verbindung

Übersicht

In diesem Unterkapitel soll der prinzipielle Aufbau einer DDE–Verbindung mit Visual Basic und mit Visual C++ zum NCDDE–Server erläutert werden.

Hinweis

In den folgenden Beispielen wird zur DDE-Kommunikation nur das Standard VisualBasic Control ”LABEL” verwendet. Eine OEM-Applikation sollte jedoch zur DDEKommunikation das Siemens-Control DCTL (siehe Kapitel 8.9.3) verwenden.

Folgende Voraussetzungen sind notwendig, damit die folgenden Beispiele funktionsfähig sind:

Entwicklungsumgebung

- wir empfehlen MS Visual Basic 4.0_16
- Um direkt vom PC aus die Beispiele auf der SINUMERIK 840D testen zu können, ist eine MPI–Anbindung notwendig und der NCDDE–Server muss für NC–Betrieb konfiguriert sein.
  Wird der NCDDE–Server ohne SINUMERIK 840D betrieben, sind nicht alle Datenzugriffe möglich.
- Der NCDDE–Server (C:\MMC2\NCDDE.EXE) muss gestartet sein (z.B. über den Explorer oder das Start–Menü).
-

8.4.1 Aufbau einer DDE–Verbindung mit Visual Basic

Bei den Standard Visual Basic Controls, die DDE – fähig sind, wie z.B.

- Label
- Textbox
- Picture

werden Link–Service und Link–Topic im Property (Attribut) “LinkTopic“ zusammengefaßt. Sie werden durch das Pipe–Zeichen “|“ getrennt (z.B. LinkTopic “ncdde|local“).

Einmaliges Lesen von Variablen

Das folgende Beispiel liest den Istwert der ersten Achse des ersten Kanals im Werkzeugkoordinatensystem **einmal** aus. Damit das folgende Beispiel funktioniert, muss der NCDDE–Server für lokalen Betrieb konfiguriert und "**NcddeServiceName = ncdde"** sein. D.h.hier erfolgt **kein** Zugriff auf den NCK. Für einmaliges Lesen muss der **LinkMode** den Wert **2** haben.

Hinweis

Für einmaliges Lesen muss der "LinkMode" den Wert 2 haben. Der Wert wird dann noch mit der Methode "LinkRequest" vom ersten Kanal angefordert.

Beispiel 8–4 Einmaliges Lesen von Variablen

```
Sub Form_Load ()
        Label1.LinkTopic = "ncdde|local"
        Label1.LinkItem="/Channel/GeometricAxis/actToolBase-
        Pos[u1,1]"
        Label1.LinkMode = 2
        Label1.LinkRequest
End Su
```

Hinweis

Wenn die Kanalbezeichnung "u1" nicht angegeben wird, wird automatisch auf denersten Kanal zugegriffen.

Aktualisierung bei Änderung

Das folgende Beispiel aktualisiert den Istwert der dritten Achse des zweiten Kanals im Maschinenkoordinatensystem automatisch (HotLink) im "Label1", d.h. der aktuelle Istwert dieser Achse wird angezeigt.

Hinweis

Für Hotlinks muss der "LinkMode" den Wert 1 haben.

Beispiel 8–5 Aktualisierung bei Änderung

```
Sub Form_Load ()
        Label1.LinkTopic = "ncdde|ncu840d"
        Label1.LinkItem="/Channel/MachineAxis/actToolBasePos[u2,3]"
        Label1.LinkMode = 1 'Hotlink
End Sub
```

Benachrichtigung bei Änderung

In diesem Beispiel benachrichtigt der NCDDE–Server die Applikation/Client, wenn sich das erste PLC–Eingangsbyte ändert (Warmlink). Dann wird automatisch die “Sub LinkNotify“ von Label1 durchlaufen. Dort muss dann der Anwender noch ein “LinkRequest“ aufrufen, um die Daten zu erhalten. Er hat dann z.B. die Möglichkeit, die Daten, bevor sie angezeigt werden, noch zu überprüfen oder umzuwandeln.

Hinweis

Für Benachrichtigung bei Änderung (Warmlink) muss der ”LinkMode” den Wert 3 haben.

Beispiel 8–6 Benachrichtigung bei Änderung

```
Sub Form_Load ( )
        Label1.LinkTopic = ”ncdde|ncu840D”
        Label1.LinkItem = ”/PLC/Input/Byte[1]” Label1.LinkMode = 3
End Sub
Sub Label1_LinkNotify ( )
        Label1.LinkRequest
End Sub
```

NC–Daten schreiben

In diesem Beispiel **schreibt** der Client den Wert “**4**“ in den R–Parameter R[1] des ersten Kanals.

Hinweis

Beim Schreiben von Daten (Poke) muss der ”LinkMode” den Wert 2 haben. Mit LinkPoke wird der Wert geschrieben.

Beispiel 8–7 NC–Daten beschreiben

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/Channel/Parameter/R[1]”
        Label1.LinkMode = 2 ’Manual
        Label1.Caption = ”4”
        Label1.LinkPoke
End Sub
```

PLC–Daten schreiben

In diesem Beispiel schreibt der Client den Wert “250“ in das Merker–Byte 5 der PLC.

Beispiel 8–8 PLC–Daten schreiben

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/PLC/Memory/Byte[5]”
        Label1.LinkMode = 2 ’Manual
        Label1.Caption = ”250”
        Label1.LinkPoke
End Sub
```

Kommando ausführen

Zum Ausführen von Kommandos, die in den Kapiteln 8.6.1, 8.6.3, 8.7 und 8.8 beschrieben werden.
Das folgende Beispiel startet einen Dateitransfer der Datei “test.mpf“ vom HMI zum NCK.

Hinweis

Beim Ausführen von Kommandos (Execute) muss der ”LinkMode” den Wert 2 haben. Mit LinkExecute wird das Kommando ausgeführt.

Beispiel 8–9 Kommando ausführen

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkMode = 2
        Label1.LinkExecute ”COPY_TO_NC(””C:\NC\test.mpf””,
        /NC/_N_MPF_DIR/N_TEST_MPF, trans)”
End Sub
```

8.4.2 Aufbau einer DDE–Verbindung mit Visual C/C++

Übersicht

Aus C/C++ heraus kann die volle Funktionalität der DDE–Schnittstelle genutzt werden. Insbesondere ist es möglich, asynchrone Aufrufe an die DDE–Schnittstelle zu richten. Unter Verwendung der OEM–Visual Basic Controls wie z.B. DCTL ist dies auch in Visual Basic möglich.

Hinweis

DDE mit C/C++ ist nur für OEM-Anwender zu empfehlen, die Erfahrung in C Programmierung unter Windows haben und nur eine bedingte oder keine Integration in die Ablaufsteuerung des OEM-Pakets wünschen.

DDE–Zugriff mit C / C++

Dieses Beispiel zeigt den Aufbau einer Hotlink (Advise)–Verbindung (mit Quittungsverkehr) zwischen

- C/C++ Programm
- Variable "/Channel/GeometricAxis/toolBaseDistToGo[1]"
- NcddeServiceName = ncdde
- NcddeMachineName = local

Die Datenänderungen der Variablen werden mit einer bei DDEML angemeldeten Callbackroutine in einer XTYP_ADVDATA Transaktion empfangen.

Beispiel 8–10 Hotlink auf C–Ebene

```
DWORD        idInst; // erzeugt mit DdeInitialize
HSZ          hszService, hszTopic, hszItem; // String Handles
HCONV        hConv; // Conversation Handle

HszService   = DdeCreateStringHandle ( idInst , ”ncdde” , NULL );
hszTopic     = DdeCreateStringHandle ( idInst , ”local” , NULL );
hszItem      = DdeCreateStringHandle ( idInst ,
               Ä”/Channel/GeometricAxis/toolBaseDistToGo[1]” , NULL );
hConv        = DdeConnect(idInst,hszService,hszTopic,NULL);
                       // Verbindungsaufbau zum Server
                       // Hotlink folgt
if ( DdeClientTransaction ( (LPBYTE)NULL , 0 , hConv , hszItem ,
             ÄCF_TEXT ,XTYP_ADVSTART|XTYP_ACKREQ , 1000 , NULL )
             Ä==TRUE) { } // Hotlink Aufbau erfolgreich
```

8.4.3 Aufbau einer DDE–Verbindung aus MS Excel

Übersicht

Unter Excel können Advise (Hotlink)–Verbindungen zur Variablenschnittstelle des NCDDE–Servers mit Zellenformeln aufgebaut werden.

Syntax in einer EXCEL–Zelle

Innerhalb einer Zelle
= NcddeServiceName|NcddeMachineName!Variablen

Anzeigen eines PLC–Bits in Excel

Das folgende Beispiel zeigt eine Advise (Hotlink)–Verbindungen zwischen einer Zelle in Excel (deutsche Version) und dem Bit 3 im Byte 9 des Datenbausteins 100.
Der Variablename lautet: "/Plc/DataBlock/Bit[c100,9.3]"
einer über den NCDDE–Server angebundenen Maschine "ncu840D".

Beispiel 8–11 Anzeigen eines PLC–Bits in MS Excel

	A			**A**
1	=ncdde|ncu840D!'/Plc/DataBlock/Bit[c100,9.3]'		1	1

Links ist die Zellenformel, rechts die resultierende, stets aktuelle Anzeige dargestellt.

8.5 Variablen–Dienst

Übersicht

Der Zugriff auf Variabeln der NC erfolgt über **OPC-Data** oder über das **DCTL-Control**. Aus Kompatibilitätsgründen besteht weiterhin die Möglichkeit, auf die Variabeln über DDE zuzugreifen.
Die Variablen–Dienste des NCDDE–Servers ermöglichen zwei Arten von Datenzugriffen:

- Einzelzugriff
- Feldzugriff

Die Variablen können im Link–Item noch durch ein zusätzliches Datenformat und evtl. auch durch einen Feldbereich gekennzeichnet werden. Dies ermöglicht es dem Anwender, die Daten vom NCDDE–Server so anzufordern, dass meistens keine Konvertierung mehr notwendig ist.

Hinweis

Ein vollständige Beschreibung der Variablen, auf die zugegriffen werden kann, entnehmen sie bitte dem **Kapitel 11** Referenz bzw. der Variablen-Online-Hilfe.

Formate der NCDDE–Variablen

Formatierungs–Anweisungen für NCDDE–Variablen werden am Ende des LinkItems eingefügt. Die interne Datenaufbereitung erlaubt die Formatierung von Festpunktwerten, Gleitpunktwerten und Texten.
Zur Beschreibung der Formatierung wird ein erweitertes ‘printf’–Format der Sprache C verwendet. Die Syntax einer NCDDE–Formatangabe ist:

Format:	““	<Params>	<‘printf’–Format>
Params:	‘!’ ‘b’	<Params>	// Konvertierung in einen Bitstring
	‘!’ ‘d’	<Params>	// d für double als 64bit floating
	‘!’ ‘l’	<Params>	// l für long als 32bit Festpunkt– //wert
	‘!’ ‘t’	<Params>	// t für text als string
	‘!’ ‘#’	<Params>	// # , Index des Variablenzu– //griffs als 32bit Festpunktwert

Der Datentyp der entsprechenden DDE–Variablen ist dem Kapitel 11 oder der “NCDDE–Variablen–Hilfe“ zu entnehmen.

Hinweis

Stimmen die Typen von Datenanwahl und ausgelesenem Variablenwert nicht überein, dann erfolgt keine automatische Anpassung des Datenformats, d.h. man bekommt falsche Daten angezeigt.

Formatierung eines Zahlenwerts

Hier wird der Istwert der **zweiten** Achse aus der NC gelesen und mit max. 11 Vorkommastellen und mit 3 Nachkommastellen dargestellt. Ohne Formatierungsanweisung bekommt man nur 3 Nachkommastellen.

Beispiel 8–12 Formatierung max. 11 Vor– und fixe 3 Nachkommastellen

```
Sub Form_Load ( )
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/Channel/MachineAxis/actToolBasePos[2]
                        Ä(””!d%11.3lf””)”
        Label1.LinkMode = 2 ’Manual
        Label1.LinkRequest
End Sub
```

Konvertierung in Hexzahl

In diesem Beispiel wird das Merker–Byte **5** gelesen und dann als zweistellige Hexadezimal–Zahl mit führenden **Nullen** angezeigt.

Beispiel 8–13 Konvertierung in Hexzahl

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/PLC/Memory/Byte[5] (””!l%02lx””)”
        Label1.LinkMode = 2 ’Manual
        Label1.LinkRequest
End Sub
```

Konvertierung in Bit–String

In diesem Beispiel wird das Merker–Byte **5** gelesen und dann als 32Bit–String ausgegeben.

Beispiel 8–14 Konvertierung in Bit–String

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/PLC/Memory/Word[5] (””!b%16.16s””)”
        Label1.LinkMode = 2 ’Manual
        Label1.LinkRequest
End Sub
```

Ergebnis: 1010101010101010101

Lesen eines Strings aus der PLC

In diesem Beispiel werden aus dem Datenbaustein **81** ab Byte **20 10** Byte gelesen und dann als Nullterminierter String ausgegeben

Beispiel 8–15 Lesen eines Strings aus der PLC

```
Sub Form_Load ()
        Label1.LinkTopic = ”ncdde|ncu840d”
        Label1.LinkItem = ”/PLC/DataBlock/Byte[c81,20,#10] (””!l%lc””)”
        Label1.LinkMode = 2 ’Manual
        Label1.LinkRequest
End Sub
```

Ergebnis: z.B. Hallo

8.5.1 Einzelzugriff

Wenn man innerhalb der Ablaufsteuerung (siehe Kapitel 7) arbeitet, ist es von Vorteil, die globale Variable “**g_chNCDDEServiceName**“ als LinkTopic zu verwenden. Sie enthält immer die in der Datei “MMC.INI“ eingetragenen **NCDDEServiceName** und den **NcddeMachineName** durch Pipe (“|“) getrennt.

Einzelzugriff auf drei Variablen

Lesen der Namen der ersten drei Maschinen–Achsen

Beispiel 8–16 Einzelzugriff auf drei Variable

```
Sub Form_Load
        achsname(0).LinkTopic = g_chNCDDEServiceName
        achsname(0).LinkItem = ”/Channel/MachineAxis/name[1]”
        achsname(0).LinkMode = 2
        achsname(0).LinkRequest
        achsname(1).LinkTopic = g_chNCDDEServiceName
        achsname(1).LinkItem = ”/Channel/MachineAxis/name[2]”
        achsname(1).LinkMode = 2
        achsname(1).LinkRequest
        achsname(2).LinkTopic = g_chNCDDEServiceName
        achsname(2).LinkItem = ”/Channel/MachineAxis/name[3]”
        achsname(2).LinkMode = 2
        achsname(2).LinkRequest
End Sub
```

PLC–Bit Zugriff

Mit folgendem LinkItem kann auf das Bit4 im Eingangsbyte 2 zugegriffen werden.
/Plc/Input/Bit[2.4]

PLC–Byte Zugriff

Mit folgendem LinkItem kann auf das Ausgangsbyte 4 zugegriffen werden.
/Plc/Output/Byte[4]

PLC–Wort Zugriff

Mit folgendem LinkItem kann auf das Merkerwort 4 zugegriffen werden.
/Plc/Memory/Word[8]
Weitere Zugriffe sind in Kapitel 12.1.5 beschrieben.

8.5.2 Feldzugriff auf Daten

Anwendung

Sollte immer verwendet werden, wenn mehrere Daten aus einem Bereich benötigt werden. Dies reduziert die Belastung des NCDDE–Servers und verbessert die Performance der eigenen Applikationen. Beispiel 8–16 ist dafür ein schlechtes Beispiel.

Hinweis

Feldzugriffe beschleunigen den Datenzugriff und verbessern auch die Geschwindigkeit des Gesamtsystems, da der Kommunikationsaufwand wesentlich reduziert wird.

Syntax

Zur Einführung soll an dieser Stelle ein kurzer Blick auf die Schreibweise von Feldbereichen geworfen werden:
Variablenname[c, u, StartIndex, [EndIndex]]

Parameter

Tabelle 8-1 Parameter für Datenzugriffe

Name	Beschreibung
Variablenname	Name der NCK/PLC-Variablen (siehe Kapitel 11)
c	**Spalten-Index** für Zugriff auf **NCK**-Variablen (siehe Kapitel 11): c steht für column (Spalte) und trifft nur auf mehrdimensionale Felder zu. Kennzeichnet beim Zugriff auf **PLC**-Datenbausteine den **Datenbaustein,** auf den zugegriffen werden soll
u	Einheiten-Index **nur** für NCK-Variablen (z.B. Kanal) u steht für unit (Einheit)
StartIndex	Index der zu lesenden Variablen. Bei Feldzugriffen ist dies der erste der zu lesenden Werte.
EndIndex (optional)	Nur für **Feldzugriffe:** gibt die Anzahl der zu lesenden Werte an.

Feldzugriff auf Achsnamen

Dieses Beispiel liest die ersten drei Achsnamen aus dem NCK. Das Ergebnis ist ein String, der diese Achsnamen z.B. X1,Y1,Z1 in der Form “X1Y1Z1“ enthält. Mit den Visual Basic Funktionen “Trim$“ und “ Mid$“ werden dann die einzelnen Achsnamen aus dem Ergebnisstring herausgelöst.

Beispiel 8–17 Feldzugriff auf Achsnamen

```
m_a_namen.LinkTopic = g_chNCDDEServiceName
m_a_namen.LinkItem = ”/Channel/MachineAxis/name[u1,1,3]”
m_a_namen.LinkMode = 2
m_a_namen.LinkRequest

’Auflösen des Wertefeldes in einzelne Werte

achsname1.Caption = Trim$(Mid$(m_a_namen.Caption,1,2))
achsname2.Caption = Trim$(Mid$(m_a_namen.Caption,4,2))
achsname3.Caption = Trim$(Mid$(m_a_namen.Caption,7,2))
```

Feldzugriff auf Achsnamen

Auslesen der Achsnamen von zwei Achsen des zweiten Kanals, beginnend bei Achse 3. Es werden die Namen der Achsen drei und vier gelesen. Es entspricht bis auf folgende Zeile Beispiel 8–16

Beispiel 8–18 Feldzugriff auf Achsname

```
...
LinkItem = ”/channel/machineaxis/name[u2,3,4]”
...
```

Feldzugriff auf PLC–Daten

Dieses Beispiel liest die drei Bytes Byte **2** bis Byte **4** des DB **8** aus der PLC: Das sind die Bytes 2,3,4. als zweistellige hexadezimale Zahlen. Die einzelnen Bytes werden danach mit den Visual Basic Funktionen “Trim$“ und “ Mid$“ aufgetrennt.

Beispiel 8–19 Feldzugriff auf PLC–Daten

```
Label1.LinkTopic = ”ncdde|ncu840d”
Label1.LinkItem = ”/PLC/Datablock/Byte[c8,2,4](””!l%02lx””)”
Label1.LinkMode = 1 ’hotlink

’Auflösen des Bytestring in einzelne Hex. Strings

byte_1 = Trim$(Mid$(Label1.Caption,1,2))
byte_2 = Trim$(Mid$(Label1.Caption,3,2))
byte_3 = Trim$(Mid$(Label1.Caption,5,2))
```

PLC–Zugriff mit Angabe der Anzahl

Dieses Beispiel liest **5** Worte des DB **8** vom Wort **2** an aus der PLC als vierstellige hexadezimale Zahlen heraus. Die einzelnen Worte sind durch “**_**“ getrennt.

Beispiel 8–20 Feldzugriff auf PLC–Daten

```
Label1.LinkItem = ”/PLC/Datablock/Word[c8,2,#5](””!l%04lx_””)”
```

Feldzugriff auf R–Parameter

Dieses Beispiel schreibt die drei R–Parameter R3, R4 und R5 mit den Werten:
R3 = 2,2
R4 = 3,5
R5 = 4,9.

Beispiel 8–21 Feldzugriff auf R–Parameter

```
Label1.LinkTopic = ”ncdde|ncu840d”
Label1.LinkItem = ”/CHANNEL/PARAMETER/R[U1,3,5]”
Label1.LinkMode = 2 ’Manual
Label1.Caption = ”:2.2:3.5:4.9”
Label1.LinkPoke
```

8.6 Dateitransfer–Dienste (Domain–Dienste)

Übersicht

Die Dateitransfer–Dienste sind für Datenübertragung zwischen den Bereichen (Domains) HMI und NCK/PLC zu verwenden.
Für den Transfer zwischen HMI und NCK/PLC stehen **IMC-File** ,**IADSI** und **IMC-Domain** zur Verfügung. Aus Kompatibilitätsgründen besteht weiterhin die Möglichkeit, den Datentransfer über DDE abzuwickeln. Hierbei stehen insgesamt fünf Befehle zur Verfügung, die in der Tabelle 8-2 zusammengestellt sind. Sie arbeiten als Hintergrundaktivität.
Ab SW–Stand 3.3 stehen erweiterte Kopierfunktionen zwischen den einzelnen Bereichen zur Verfügung. Diese eignen sich besonders zum Editieren von Programmen in der NC. Die neuen Funktionen sind im Kapitel 9.6.2 näher beschrieben.

Tabelle 8-2 Parameter für Datenzugriffe

Name	**Beschreibung**
COPY_FROM_NC	Transfer vom NCK zur HMI
COPY_FROM_NC_BINA RY	Transfer vom NCK (PLC) zur HMI
COPY_TO_NC	Transfer von HMI zum NCK
COPY_TO_NC_BINARY	Transfer von HMI zum NCK (PLC)
MAP_ACC_NC	Laden von ACC-Dateien aus NC-Kern und Aufbereitung für DDE-Schnittstelle

Der Zustand einer Datenübertragung (Transfers) läßt sich über eine Status–Variable verfolgen.

8.6.1 Dateiübertragung zwischen HMI und NC/PLC

Beschreibung

Diese Funktionen erlauben es dem Anwender, Daten/ Dateien zwischen HMI und NCK/PLC zu übertragen

Anwendung

Diese Funktionen eignen sich für die Übertragung von Teileprogrammen, Werkstücken in den NCK und von S7 und C–Programmen in die PLC.
Mit den Funktionen **ohne** die Erweiterung "BINARY" können Dateien, z.B. Teileprogramme in die NC übertragen werden. Der NCDDE–Server ergänzt die Daten, die übertragen werden sollen, um einen Bausteinkopf, der die Bausteingröße und das Bausteindatum enthält, sowie um den Bereichspfad.

Hinweis

- Wird zur Übertragung von Daten zum NCK verwendet.
- Kann nicht zur Datenübertragung zur PLC verwendet werden, da immer ein NCBausteinkopf zum Datenstrom hinzugefügt wird.

BINARY–Funktionen

Mit den Funktionen **mit** der Erweiterung "BINARY" können Dateien, z.B. Teileprogramme in den NCK bertragen werden. Der NCDDE–Server überträgt die Daten ohne NC–Bausteinkopf.

Hinweis

- Kann zur Datenübertragung zur PLC und zum NCK verwendet werden.
- PLC-Bausteine werden immer in das passive Dateisystem der PLC übertragen. Sie sind dann aber noch nicht aktiv. Die passiven Bausteine müssen dann noch aktiviert werden.

Syntax

Die Kopierfunktionen werden als String in der folgenden Form geschrieben:
COPY_FROM_NC (WinFile,NcFile,TransferState)
COPY_TO_NC (WinFile,NcFile,TransferState)
COPY_FROM_NC_BINARY (WinFile,NcFile,TransferState)
COPY_TO_NC_BINARY (WinFile,NcFile,TransferState)

Parameter

Tabelle 8-3 Parameter für Datenzugriffe

Name	Beschreibung
WinFile	Quelle oder Ziel der Information im HMI–Bereich:
NcFile	Dateiname für die NCK/PLC Umgebung
TransferState	Variable kennzeichnet Verbindungszustand

Parameter WinFile

Beschreibt die Quelle oder das Ziel der Information auf der HMI–Seite. Das erste Zeichen gibt an, um welchen Typ es sich dabei handelt.
Standardmäßig gilt dieser Parameter als Dateiname in der WINDOWS–Umgebung, der dann immer Laufwerksbezeichnung, Pfad– und Dateinamen umfassen sollte. z. B. **"C:\NC\test.MPF"**.

Piping mit dem Parameter WinFile

Wenn WinFile mit einem @–Zeichen beginnt, dann wird der Parameter als Pipe–Name interpretiert. In Verbindung mit COPY_TO_NC kann damit der Dienst "Kopieren über Pipes" ausgeführt werden.

Hinweis

Eignet sich zum Schreiben und Lesen von Blöcken bis 500 Byte Größe. Größere Blöcke lehnt der NCDDE-Server ab.

Beim Transfer zum NCK/PLC (download) füllen DDE–Pokes die Pipeline und sorgen damit für direkte Übertragung zum NCK/PLC. Ein leeres Poke zeigt das Ende des Transfers an.
Beim Transfer vom NCK/PLC (upload) leeren DDE–Requests die Pipeline, die für den laufenden Transfer gefüllt wird. Liefert ein Request leere Daten, dann zeigt dies das Ende des Transfers an.

Shared Memory - Zugriff mit dem Parameter WinFile

Wenn WinFile mit einem #–Zeichen beginnt, gefolgt von einer hexadezimalen Zahl, wird dies als WINDOWS Shared memory interpretiert, dss dem Global Heap zugeordnet ist. Der mit der Windows Funktion Global/Alloc allokierte Speicher muss mit der folgenden Struktur initialisiert weden. Im Anschluß an diesen Header folgen die Nutzdaten. In Beispiel 8–22 ist die Verwendung von Visual Basic aus angezeigt.

Beispiel 8–22 Shared Memory - Zugriff mit WinFile

```
struct NCDDE_DOMAINMAP_HEADER {
    unsigned short handle;         // handle des Puffers (HGLOBAL) wird vom client
                                   // vorbesetzt)
    unsigned short header_size;    // laenge des headers (wird vom client vorbesetzt)
    unsigned long shared_size;     // nutzbare laenge des datenbereichs (wird vom client
                                   // vorbesetzt)
    unsigned long fill_count;      // anzahl gueltiger bytes im Datenbereich (wird beim
                                   // download vom client vorbesetzt und beim upload vom
                                   // server gesetzt)
    unsigned long state;           // entspricht der transferzustandsvariablen des
                                   // transferkommandos
                                   // < 100: der transfer läuft, “state” zeigt wieviele prozente
                                   // des transfers abgewickelt wurden
                                   //  ==100: transfer erfolgreich abgeschlossen
                                   // > 100: der transfer wurde mit fehler abgebrochen,
                                   // ”state” zeigt den ncdde fehlercode
                                   // (wird vom server gesetzt)
    unsigned long file_mod_time;// file modification time wert:
                                   // 0 bedeutet current time (wird beim download vom client
                                   // vorbesetzt und beim ipload vom server gesetzt)
    unsigned long server_private;// serverspezifisches Datum (wird vom server gesetzt)
    unsigned long client_private; // clientspezifisches Datum (wird vom client gesetzt)
    unsigned long magic;           // signatur fuer eine zusätzliche typpruefung,  wert immer
                                   // NCDDE_MAGIC = 0xF6F7F8F9 (wird vom client
                                   // vorbesetzt(
};
```

Parameter NcFile

Der Parameter “NcFile“ ist der Dateiname für die NCK/PLC Umgebung. Er setzt sich zusammen aus einem projektierbaren Pfadnamensanteil, der zur Adressierung der betroffenen NCK erforderlich ist, und einem Domainpfad der NCK–Umgebung.
Domains im NCK werden über den NCDDE–Server mit Hilfe eines NC–Dateinamens adressiert.

/NC	Bereichsangabe PLC oder NCK
/_N_MPF_DIR	Pfadangabe für NC
/_N_WS03_MPF	Name der Datei

Parameter TransferState

Der Parameter “TransferState“ ist der Name einer NCDDE lokalen Variablen (Variablentyp: Fixed), die zur Rückmeldung des Zustandes des im Hintergrund ablaufenden Transfers verwendet werden soll. Diese Variable wird gegebenenfalls auch vom NCDDE–Server erzeugt.
Die Variable TransferState kennzeichnet den Übertragungszustand:

Tabelle 8-4 Kennzeichnung des Zustandes

Zustand des Transfers	Wert	Bedeutung
Transfer wird gestartet	0	Mit der CNC läuft das Open Protokoll (Öffnen der Datei)
Transfer läuft	1 bis 98	Transfer läuft, der Wert zeigt etwa den Prozentanteil der bereits transferierten Daten (siehe Hinweis)
Transfer wird beendet	99	Mit der CNC läuft das Close Protokoll (Schließen der Datei)
Transfer erfolgreich	100	Der Auftrag wurde fehlerfrei ausgeführt
Transfer gestoppt mit Fehlernummer	>100	Transfer gestoppt, TransferState enthält den gemeldeten Fehlercode. Dieser entspricht dem Wert der Variablen LastError (Kapitel 11.7).

Der Wertebereich wurde so gewählt, dass Werte <=100 Normalbedingungen und alle übrigen Werte Fehlerbedingungen bedeuten.

Hinweis

Wenn die Variable einen Wert von 1 bis 99 hat, ist mit ihr kein weiterer Dateitransfer möglich.

Datenübertragung stoppen

Um eine laufende Datenübertragung zu stoppen, muss die Transfervariable mit einem ordentlichen Fehlerwert beschrieben werden. D.h. jedes Byte der als “LONG“ (4 Byte) definierten Transfervariable muss einen Wert **ungleich** 0 haben .

Beispiel für einen ordentlichen Fehlerwert: **16909060**

Visualisierung

Zur Visualisierung des Übertragungszustandes kann die Variable TransferState per Advise/Hotlink mit einer Balkenanzeige o.ä. verbunden werden.

Hinweis

Da bei einer Übertragung im BINARY-Modus und bei der Datenübertragung mit Pipes Informationen über die Bausteingröße fehlen, kann die Variable Transfer-State nicht mit dem aktuellen Prozentsatz der übertragenen Daten versorgt werden: Sie steht dann konstant auf 50%

Bei sehr kurzen Dateien kann die Anzeige sofort von 1 auf 99 springen. Dies ist aber ein prinzipielles Problem bei Hotlinks und liegt daran, dass der Client/Applikation die Daten nicht schnell genug vom NCDDE–Server abholt.

Upload eines Teileprogrammes

Das folgende Beispiel kopiert das Teileprogramm “BSP.MPF“ in die Datei “test.mpf“ im Verzeichnis “C:\NC“. Die Datei “test.mpf“ wird neu angelegt. Das Teileprogramm “BSP.MPF“ muss im NCK vorhanden sein.

Beispiel 8–23 Upload eines Teileprogrammes

```
Sub Form_Load ()
        Label1.LinkTopic = ”NCDDE|ncu840d”
        Label1.LinkMode = 2
        Label1.LinkExecute  ”COPY_FROM_NC (C:\nc\test.mpf,
                                 Ä/NC/_N_MPF_DIR/_N_BSP_MPF,trans)”
End Sub
```

Download eines Teileprogramms

Das folgende Beispiel kopiert eine Datei mit dem Namen “test.mpf“ aus dem Verzeichnis “C:\NC“ zum NCK in das “_N_MPF_DIR“ Directory. Das Teileprogramm in der NC heißt “**BSP.MPF**“

Beispiel 8–26 Download eines Teileprogramms

```
Sub Form_Load ()
        Label1.LinkTopic = ”NCDDE|ncu840d”
        Label1.LinkMode = 2
        Label1.LinkExecute ”COPY_TO_NC(C:\NC\test.MPF,
                                Ä/NC/_N_MPF_DIR/_N_BSP_MPF,trans)”
End Sub
```

Download eines Teileprogramms mit Piping

Das folgende Beispiel zeigt die Nutzung des **Pipe**–Mechanismus: Die Datei PIPE1.MPF wird im NCK angelegt und der NC–Satz “G01 F11111 X5555“ hineingeschrieben.

Beispiel 8–27 Download eines Teileprogramms mit Piping

```
Sub Form_Load ()
        'Pipe starten
        Label1.LinkTopic = "NCDDE|ncu840d"
        Label1.LinkMode = 2
        Label1.LinkExecute "COPY_TO_NC(@pipe,
                                Ä/NC/_N_MPF_DIR/_N_PIPE1_MPF,trans)"
        'Pipe beschreiben
        Label2.LinkTopic = "NCDDE|NCU840D"
        Label2.LinkMode = 2
        Label2.LinkItem = "@pipe"
        Label2.Caption = "G01 F11111 X5555"
        Label2.LinkPoke
        'beenden der Pipe
        Label2.Caption = ""
        Label2.LinkPoke
End Sub
```

Download eines S7–Bausteines zur PLC

Übertragen des Bausteins “**OB1.PLC**“ zur PLC ins passive Dateisystem

---

Hinweis

PLC--Bausteine werden immer in das passive Dateisystem der PLC übertragen. Sie sind dann aber noch nicht aktiv. Die passiven Bausteine müssen dann noch aktiviert werden (vgl. Beispiel 8-36).

Beispiel 8–28 Download eines S7–Bausteines zur PLC

```
Label1.LinkItem = "ncdde|ncu840d"
Label1.LinkMode = 2
Label1.LinkExecute "COPY_TO_NC_BINARY(C:\TMP\OB1.PLC,
                    Ä/PLC/_0800001P, trans)"
```

8.6.2 Erweiterte Datenübertragung zwischen HMI und NC/PLC

Beschreibung

Diese Funktionen erlauben es dem Anwender, Daten zwischen NCK/PLC und HMI zu übertragen.

Anwendung

Diese Funktionen eignen sich besonders für die Übertragung von einzelnen Sätzen, Programmteilen und auch zum Editieren von Teileprogrammen auf der NC.

Hinweis

Der Unterschied zwischen der ”normalen” und den ”BINARY”-Varianten der Funktionen ist in Kapitel 8.6.1 beschrieben.

Syntax

Die erweiterten Kopierfunktionen werden als String in der folgenden Form geschrieben:

COPY_FROM_NC	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_FROM_NC(_BINARY)	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_TO_NC	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_TO_NC(_BINARY)	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)

Parameter

Die Parameter sind in der Tabelle beschrieben.

Tabelle 8-5 Parameter der Befehle COPY_TO/FROM_NC

Name	Beschreibung
WinFile	Quelle oder Ziel der Information im HMI–Bereich:
NcFile	Dateiname für die NCK/PLC Umgebung
seekPos	Seek–Zeiger: Startpunkt des Kopiervorgangs Kennung B für Block oder C für Zeichen
seekLen	Fenstergröße: Bereich, der übertragen wird Kennung B für Block oder C für Zeichen
compareString	Suchstring, bis zu 32 Zeichen lang
skipCount	Anzahl der gefundenen Suchstrings, die überlesen werden sollen

Die Kommandos kehren erst nach vollständiger Bearbeitung aller Teilkommandos zurück. Fehler, die während der Ausführung des Kommandos erkannt werden, lassen sich anhand der Variablen LastError analysieren.

Das folgende Beispiel zeigt eine typische Anwendung der erweiterten Befehle.

Filetransfer eines Programmteils

Filetransfer der ersten **1024** Byte des Teileprogramms “**TP1.MPF**“ in die Datei “test.dat“ im Verzeichnis “C:\NC“

Beispiel 8–29 Filetransfer eines Programmteils

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute”COPY_FROM_NC(C:\NC\test.dat,
                                Ä/NC/_N_MPF_DIR/_N_TP1_MPF,1 ,1024 , , 0
)”
End Sub
```

Transfer einzelner Sätze

Pipe–Transfer der Sätze 2 bis 4 in das Teileprogramm X.MPF. Existierende Sätze werden überschrieben.

Beispiel 8–30 Transfer einzelner Sätze

```
Sub Form_Load ( )
          Label1.LinkTopic = ”NCDDE|NCU840D”
          Label1.LinkMode = 2
          Label1.LinkExecute ” COPY_TO_NC_BINARY ( @xpipe ,
                    Ä/NC/_N_MPF_DIR/_N_X_MPF , B2 , 3 , , 0 )”
End Sub
```

Transfer eines Satzes

Text–Transfer (Textlänge max. 200 Byte) in den 2. Satz des Teileprogramms TEST.MPF. Der zweite Satz wird überschrieben.

Beispiel 8–31 Transfer eines Satzes

```
Sub Form_Load ( )
          Label1.LinkTopic = ”NCDDE|NCU840D”
          Label1.LinkMode = 2
          Label1.LinkExecute ”COPY_TO_NC ( ””!Dies wird der Inhalt des
          2.Satzes”” , /NC/_N_MPF_DIR/_N_TEST_MPF, B2 ,1 , , 0 )”
End Sub
```

8.6.3 MAP–Funktionen zwischen Domains

MAP_ACC_NC

Beschreibung
Diese Funktion erlaubt es dem Anwender, Globale–Anwender–Daten (GUDs) und Maschinendaten des NCK dem NCDDE–Server bekannt zu machen. Diese Daten sind in Dateien mit der Endung ACC gespeichert. Dateien mit der Ergänzung ACC liegen im NCK und enthalten Zugriffsbeschreibungen der Variablen.

Anwendung
Mit dem Befehl MAP_ACC_NC ist es möglich, ACC–Dateien aus dem NCK zu lesen und für die NCDDE–Schnittstelle aufzubereiten. D.h. die entsprechenden Verbindungen zu diesen Daten werden im NCDDE– Server angelegt / bekannt gemacht.

Hinweis

Ermöglicht es Anwendern, dem NCDDE-Server neue Daten des NCK bekannt zu machen. Auf diese Variablen/Daten des NCK könnte ansonsten nicht zugegriffen werden.

Der Befehl verhält sich wie der Befehl COPY_FROM_NC mit der Erweiterung, dass er die aus der ACC–Datei übertragene Information noch decodiert und in einer Weise aufbereitet, dass sie an der DDE–Schnittstelle darstellbar ist.

Syntax
Der Aufruf erfolgt in folgender Form:
MAP_ACC_NC
Ä(WinFile, NcFile, TransferState, Area, DataBlock, Timeout, Prefix)

Parameter
Eine nähere Beschreibung der Parameter folgt in der Tabelle 8-6. Die ersten drei Parameter entsprechen denen der anderen Domain–Dienste (siehe 9.6.1). Sie werden der Vollständigkeit halber hier noch mit aufgeführt.

Tabelle 8-6 Parameter des Befehls MAP_ACC_NC

Name	Beschreibung
WinFile	Quelle oder Ziel der Information im HMI–Bereich
NcFile	Dateiname für die NCK/PLC–Umgebung
TransferState	Variable kennzeichnet Verbindungszustand
Area	Bereichsadresse der ACC–Daten, beschrieben im Kapitel 11.1.1, Tab.11–1. Hier die komplette Übersicht: **Bereich** – **Bereichsadresse (Area)** NCK – 0 BAG – 1 Kanal – 2 Achse – 3 Werkzeug/Magazin – 4 Vorschub–Antrieb – 5 Hauptspindel–Antrieb – 6 reserviert – 7
DataBlock	Baustein für den Variablendienst: Numerischer HEX–Wert von 00 bis FF, beschrieben im Kapitel 11.3.1, zum Beispiel (Auszug): **Baustein–Bezeichnung** – **Nummer (DataBlock)** Systemdaten (Y) – 10 globale Anwenderdaten (GUD) – 17 Werkzeugdaten für OEM (TU) – 24 Magazinverzeichnis (TMV) – 2B
Timeout	Zeitüberwachung der Transaktion NCK–HMI in Sekunden
Prefix	Beliebiger String, der vor die ACC–Variable gestellt wird.

Hinweis

Wenn es sich beim Parameter WinFile um eine Datei mit der Endung .NSK handelt, dann erzeugt der Domain-Dienst neben einer ACC-Datei noch eine NSK-Datei, in der die zugeordneten LINK-Kommandos stehen.

ACC–Dateien

/NC/_N_NCK_GD2_ACC	;	globale NCK-Anwendervariable MGUD
/NC/_N_CH02_GUD_ACC	;	globale Anwendervariable im 2. Kanal
/NC/_N_AX_SEA_ACC	;	axiale Settingdaten
/NC/_N_CH_TEA_ACC	;	NC-Maschinendaten Kanal

Anlegen von Verbindungen für Antriebs–Maschinendaten

MAP_ACC_NC	Befehlskopf
L:\MMC2\NCMDACC.NSK	Dateiname in der WINDOWS-Umgebung
/NC/_N_VS_DIR/_N_VS_TEA_ACC	NC–Domain
trans	Variable TransferState
5	Area, hier Ziffer 5 für Bereichsadresse Vorschub–Antrieb
7F	DataBlock, hier Adresse 7F für den Baustein Servicewerte Antriebe
10	Zeitüberwachung, hier 10 Sekunden
/ACC/driveVSA/MD/	Prefix, hier die Zeichenkette mit der später auf die Daten zugegriffen werden soll

Beispiel 8–32 Anlegen von Verbindungen für Antriebs–Maschinendaten

```
Sub Form_Load ()
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ”MAP_ACC_NC (L:\MMC2\NCMDACC.NSK,
        /NC/_N_VS_DIR/_N_VS_TEA_ACC,trans,5,7F,10,/ACC/driveVSA/MD/)”
End Sub
```

Zugriff auf erzeugte Verbindungen

Der Zugriff auf einen im Beispiel vorher erzeugten Link mit folgenden Komponenten:

/ACC/driveVSA/MD/	Prefix vom vorherigen Aufruf des MAP–Kommandos
$MD_TORQUE_THRESHOLD_X[1]	Name des Maschinendatums, begonnen mit $.

Beispiel 8–33 Zugriff auf erzeugte Verbindungen

```
Sub Form_Load ()
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkItem= ”/ACC/driveVSA/MD/$MD_TORQUE_THRESHOLD_X[1]”
        Label1.LinkRequest
End Sub
```

Zugriff auf globale Anwendervariable

Der Zugriff auf globale Anwendervariable ist in Kapitel 9.13 beschrieben

Beispiele einigerMAP_ACC_NC–Kommandos

Zu beachten ist bei diesen Beispielen, dass zwischen dem Parameter WinFile und NcFile nach dem Komma ein Leerzeichen folgen muss.

Beispiel 8–34 Beispiele von MAP_ACC_NC–Kommandos

```
Alle Maschinendaten:
MAP_ACC_NC(c:\tmp\c.nsk, /NC/_N_COMPLETE_TEA_ACC,trans,0,1A,10,/MD/)

Alle NCK-Maschinendaten:
MAP_ACC_NC(c:\tmp\nc.nsk, /NC/_N_NC_TEA_ACC, trans,0,1A,10,/NC/)

Kanal-Maschinendaten des Kanals 1:
MAP_ACC_NC(c:\tmp\ch1.nsk, /NC/_N_CH1_TEA_ACC, trans,2,1A,10,/CH1/)

Alle achspezifischen Maschinendaten:
MAP_ACC_NC(c:\tmp\ax.nsk, /NC/_N_AX_TEA_ACC, trans,3,1A,10,/AX/)

Alle NC-globalen Settingdaten:
MAP_ACC_NC(c:\tmp\sea.nsk, /NC/_N_NC_SEA_ACC, trans,0,16,10,/SEA/)

Alle achsspezifischen Settingdaten:
MAP_ACC_NC(c:\tmp\axs.nsk, /NC/_N_AX_SEA_ACC, trans,3,16,10,/AXSEA/)

Alle NC-globalen Anwenderdaten:
MAP_ACC_NC(c:\tmp\gud.nsk, /NC/_N_NC_GUD_ACC, trans,0,17,10,/GUD/)

Alle kanalspezifischen Anwenderdaten:
MAP_ACC_NC(c:\tmp\gud.nsk, /NC/_N_CH_GUD_ACC, trans,2,17,10,/GUD/)

Alle NC-globalen Anwenderdaten 1 (=SGUD):
MAP_ACC_NC(c:\tmp\gd1.nsk, /NC/_N_NC_GD1_ACC, trans,0,17,10,/GUD1/)

Alle kanalspezifischen Anwenderdaten 1 (=SGUD):
MAP_ACC_NC(c:\tmp\gd1.nsk, /NC/_N_CH_GD1_ACC, trans,2,17,10,/GUD1/)

Alle NC-globalen Anwenderdaten 2 (=MGUD):
MAP_ACC_NC(c:\tmp\gd2.nsk, /NC/_N_NC_GD2_ACC, trans,0,2D,10,/GUD2/)

Alle kanalspezifischen Anwenderdaten 2 (=MGUD):
MAP_ACC_NC(c:\tmp\gd2.nsk, /NC/_N_CH_GD2_ACC, trans,2,2D,10,/MGUD/)

Alle NC-globalen Anwenderdaten 3 (=UGUD):
MAP_ACC_NC(c:\tmp\gd3.nsk, /NC/_N_NC_GD3_ACC, trans,0,2E,10,/GUD3/)

Alle NC-globalen Anwenderdaten 4 (=GUD4):
MAP_ACC_NC(c:\tmp\gd4.nsk, /NC/_N_NC_GD4_ACC, trans,0,2F,10,/GUD4/)

Alle NC-globalen Anwenderdaten 5 (=GUD5):
MAP_ACC_NC(c:\tmp\gd5.nsk, /NC/_N_NC_GD5_ACC, trans,0,30,10,/GUD5/)

Alle NC-globalen Anwenderdaten 6 (=GUD6):
MAP_ACC_NC(c:\tmp\gd6.nsk, /NC/_N_NC_GD6_ACC, trans,0,31,10,/GUD6/)

Alle NC-globalen Anwenderdaten 7 (=GUD7):
MAP_ACC_NC(c:\tmp\gd7.nsk, /NC/_N_NC_GD7_ACC, trans,0,32,10,/GUD7/)

Alle NC-globalen Anwenderdaten 8 (=GUD8):
MAP_ACC_NC(c:\tmp\gd8.nsk, /NC/_N_NC_GD8_ACC, trans,0,33,10,/GUD8/)

Alle NC-globalen Anwenderdaten 9 (=GUD9):
MAP_ACC_NC(c:\tmp\gd9.nsk, /NC/_N_NC_GD9_ACC, trans,0,34,10,/GUD9/)
```

8.7 PI-Dienste

Übersicht

Um Programminstanz(PI)–Dienste auf der NC/PLC auszuführen, steht **IMC-Command** zur Verfügung. Aus Kompatibilitätsgründen besteht weiterhin die Möglichkeit, Aufträge über DDE an den NCK und die PLC zu ermitteln. Eine Zusammenstellung der PI–Dienste gibt es in **PI.hlp**.
Die PI–Dienste des NCDDE–Servers umfassen:

PI_START	Auftrag an NCK, ein Kommando auszuführen
PI_START_BINARY	Auftrag an PLC, ein Kommando auszuführen
PI_STOP	Auftrag an NCK, die Ausführung eines Kom–mandos anzuhalten
PI_STOP_BINARY	Auftrag an PLC, die Ausführung eines Kom–mandos anzuhalten
PI_RESUME	Auftrag an NCK, ein angehaltenes Kommando fortzuführen
PI_RESUME_BINARY	Auftrag an PLC, ein angehaltenes Kommando fortzuführen

PI_START(_BINARY)

Beschreibung
Diese Funktion erlaubt es dem Anwender, einen Auftrag vom HMI an den NCK abzusetzen.

Anwendung
Diese Funktionen eignen sich für das Starten von Aufträgen im NCK.
Der nichtbinäre Transfer eignet sich für die Übertragung zur NCK.
Der binäre Transfer eignet sich zur Übertragung von Daten zur PLC, zum NCK und den Antrieben.

Syntax
Die Kommandozeile zum Aufruf eines PI–Dienstes ist folgendermaßen strukturiert:
PI_START(Server–Name, Parameter 1, Parameter 2 ... Parameter n, PI–Name)
PI_START_BINARY (Server–Name, Parameter , PI–Name)
Der PI–Name für NCK beginnt mit _N_ , gefolgt von 6 Zeichen. Für die PLC gelten leicht veränderte Konventionen.

Parameter
Die Parameter sind in der Online–Hilfe ausführlich beschrieben, da sie in Abhängigkeit vom jeweiligen PI–Dienst sehr unterschiedliche Bedeutung haben können.

Teileprogramm anwählen

Dieses Beispiel zeigt, wie der PI–Auftrag “**SELECT**“ (Programm wählen zum Abarbeiten für einen Kanal) das Teileprogramm “**BSP.MPF**“ anwählt.
Dabei ist zu beachten, dass in diesem Kommando der Bereichspfad, nicht der NC–Dateipfad einzutragen ist.

Beispiel 8–35 Teileprogramm anwählen

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ”PI_START(/NC,201,/_N_MPF_DIR/_N_BSP_MPF,
        _N_SELECT)”
End Sub
```

OB 1 aktivieren

OB1 aktivieren, der bereits im passiven Dateisystem der PLC steht:

Beispiel 8–36 OB 1 aktivieren

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ”PI_START_BINARY( /PLC,
        ””@1d1@1d0@@0800001P””,_INSE)”
End Sub
```

Teileprogramm anwählen stoppen

Dieses Beispiel zeigt, wie der PI–Auftrag “SELECT“ (Programm wählen zum Abarbeiten in einem Kanal) für das Teileprogramm “BSP.MPF“ **gestoppt** wird.

Beispiel 8–37 Teileprogramm anwählen **stoppen**

```
Sub Form_Load ( )
       Label1.LinkTopic = ”NCDDE|NCU840D”
       Label1.LinkMode = 2
       Label1.LinkExecute ”PI_STOP(/NC,201,/_N_MPF_DIR/
                           Ä_N_BSP_MPF, _N_SELECT)”
End Sub
```

OB 1 aktivieren stoppen

Beispiel 8–38 OB 1 aktivieren **stoppen**

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ”PI_STOP_BINARY( /PLC,
        Ä””@1d1@1d0@@0800001P””, _INSE)”
End Sub
```

8.8 Weitere Kommandos des NCDDE–Servers

Übersicht

Weitere NCDDE–Server–Kommandos sind in der Tabelle 8-7 zusammengefasst:

Tabelle 8-7

Kommando	Bedeutung
NEW	Erzeugen lokaler Variablen
FREE	Löschen von Variablen
ANIMATE	Ständiges Ändern einer “lokalen“ Variablen
CALL	Ausführung von NCDDE–Kommandos in Dateien
PLC_MEMORYRESET	Urlöschen der PLC

NEW

Beschreibung
Erzeugt eine lokale/interne Variable im NCDDE–Server, auf die dann zugegriffen werden kann.

Anwendung
Mit dem Kommando “NEW“ wird eine NCDDE–Server lokale/interne Variable erzeugt. Beim Zugriff auf diese Variable läuft keine Kommunikation mit dem NCK ab. Existiert eine Variable des Namens VarName bereits, dann wird diese zuvor gelöscht (Verhalten wie das Kommando “FREE“ Kapitel 8.8).

Syntax
NEW (VarName , Wert)

Parameter

Tabelle 8-8 Parameter von NEW

Parameter	**Syntax**	**Bedeutung**
VarName	<String>	Name der Variablen, die erzeugt werden soll
Wert	<Parameter>	Initialisierungswert für die Variable

Interne Variable erzeugen
Erzeugt die Variable “test“ im NCDDE–Server und initialisiert sie mit dem Wert **10.0**.

Beispiel 8–39 Interne Variable erzeugen

```
Sub Form_Load ( )
Label1.LinkTopic = ”NCDDE|NCU840D”
Label1.LinkMode = 2
Label1.LinkExecute ” NEW ( test , 10.0 )”
End Sub
```

FREE

Beschreibung
Löschen einer Variable im NCDDE–Server

Anwendung
Mit dem Kommando “FREE“ werden mit den Kommandos “NEW“, “LINK“, erzeugte Variable gelöscht. Wenn die Variable gerade von einem Dateitransfer–Dienst (Kapitel 9.6.1) als Statusvariable verwendet wird, dann wird das Kommando “FREE“ zurückgewiesen. Bestehen Advise Links (Hotlinks) auf diese Variable, dann werden diese abgebaut. Sonstige Transaktionen mit der CNC werden abgebrochen.

Syntax
FREE (VarName)

Parameter

Tabelle 8-9 Parameter von FREE

Parameter	Syntax	Bedeutung
VarName	<String>	Name der Variablen, die gelöscht werden soll

Interne Variable löschen
Löscht die Variable “**test**“ im NCDDE–Server

Beispiel 8–40 Interne Variable löschen

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ” FREE( test )”
End Sub
```

ANIMATE

Beschreibung
Veranlaßt den NCDDE–Server, eine mit “NEW“ angelegte lokale Variable ständig zu verändern. Zahlenwerte werden im Takt von ungefähr einer Sekunde inkrementiert.

Anwendung
Zum Testen der eigenen Applikation zu verwenden.

Syntax
Animate (VarName)

Parameter

Tabelle 8-10 Parameter von Animate

Parameter	Syntax	Bedeutung
VarName	<String>	Name der Variablen, die verändert werden soll

Interne Variable ändern
Verändert die Variable “**test**“ im NCDDE–Server ständig.

Beispiel 8–41 Interne Variable ändern

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ” ANIMATE( test )”
End Sub
```

CALL

Beschreibung
Dient der Interpretation einer Kommandodatei

Anwendung
Mit dem Kommando CALL werden in Dateien abgelegte NCDDE–Kommandos ausgeführt. Jede Zeile der Datei wird dem NCDDE–Server als Kommando übergeben. Die Datei darf Kommentar und Leerzeilen enthalten. Für NCDDE–Kommandodateien ist einheitlich die Namenserweiterung .NSK zu verwenden.

Hinweis

Erlaubt es dem Anwender, den NCDDE-Server mit dem von ihm benötigten Variablen anzulegen.

Syntax
CALL (FileName)

Parameter

Tabelle 8-11 Parameter von CALL

Parameter	Syntax	Bedeutung
VarName	<String>	Name der NCDDE–Kommandodatei

Beispiel siehe Datei “\MMC2\ NCDDE311.NSK“

PLC_MEMORYRESET

Beschreibung
Mit dem Kommando PLC_MEMORYRESET des NCDDE–Servers läßt sich ein Urlöschen der PLC durchführen. Als Bereichsadresse ist /PLC anzugeben.

Anwendung
Urlöschen der PLC
Syntax
PLC_MEMORYRESET(BereichsAdr)

Parameter

Tabelle 8-12 Parameter von PLC_MEMORYRESET

Parameter	Syntax	Bedeutung
BereichsAdr	<String>	Bereichsadresse

Urlöschen der PLC
Urlöschen der PLC; die PLC muss allerdings vorher gestoppt werden

Beispiel 8–42 Urlöschen der PLC

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|NCU840D”
        Label1.LinkMode = 2
        Label1.LinkExecute ”PLC_MEMORYRESET(/PLC)”
End Sub
```

8.9 OEM–VisualBasic Controls (OCX–Dateien)

Übersicht

Mit diesen Controls werden einige Unzulänglichkeiten der DDE–Kommunikation mit Standard–Controls von Visual Basic überwunden.

Hinweis

Bei der Entwicklung einer OEM-Applikation wird empfohlen, diese OEM-Visual Basic Controls für Zugriffe auf den NCDDE-Server zu verwenden

---

Standardcontrols wie Label und Textfeld bieten DDE–Kommunikation an. Diese hat jedoch einige Unzulänglichkeiten:

- Events gehen verloren.
  Es ist nicht sichergestellt, dass die change–procedure im VB–Programm angesprungen wird, wenn sich bei Linkmode = 1 der Wert einer DDE–Variable ändert. (Abhilfe nur durch Pollen des Wertes über “Timercontrol“ möglich.)
- DDE Funktionalität kann nicht geschachtelt werden.
  In einer DDE–change–procedure kann keine weitere DDE–Funktionalität der Controls aktiviert werden (Abhilfe wieder nur über Timer o.ä. möglich.)
- Es sind nur synchrone Transaktionen realisiert
  Die Reaktionszeiten beim Einrichten von Hotlinks und bei Requests sind recht lang. Dies gilt besonders, wenn diese Aktionen über mehrere CPUs laufen (NCK, PLC).
- Ressourcenverbrauch ist hoch
  Für jede Controlinstanz, die DDE verwendet, wird eine DDE–Vebindung eingerichtet. Jede Verbindung verbraucht 2 Windowhandles und damit knappe User–Ressourcen.
- LastError bei NCDDE wird nicht komfortabel behandelt
  Bei Kommunikation zum NCK über NCDDE stellt dieser die DDE–Variable “LastError“ zur Feindiagnose von Fehlern zur Verfügung. Diese Variable ist DDE–konversationsspezifisch und nur gültig, falls der DDE–Return “DDE_FNOTPROCESSED“ ist.

8.9.1 Datei DDECTL.VBX

Durch die Umstellung von VB–Applikationen von 16Bit auf 32Bit, entfällt DDECTL.VBX (siehe auch Kapitel “Portierung der VB–Applikationen von 16Bit auf 32Bit“).

8.9.2 Datei DCTL.VBX

Durch die Umstellung von VB–Applikationen von 16Bit auf 32Bit, wurde DCTL.VBX zu DCTL.OCX (siehe folgendes Kapitel und auch Kapitel "Portierung der VB–Applikationen von 16Bit auf 32Bit").

8.9.3 Datei DCTL.OCX

Übersicht

Das Visual Basic Control DCTL.OCX ist ein grafisches Control mit erweiterter DDE–Funktionalität. Es ähnelt dem Standard–Control Label, bietet darüber hinaus aber mehrere Vorteile:

- Minimaler Verbrauch von WINDOWS–Ressourcen:
  DDE Request, DDE Poke und DDE Execute benötigen ihre Ressourcen nur temporär. Alle DDE–Hotlinks eines WINDOWS–Prozesses mit dem Control DCTL.OCX benötigen zusammen nur ein WINDOWS–Handle.
- Enge Anbindung an den NCDDE–Server:
  Es liefert z.B. den LastError–Wert bei erfolglosen DDE–Transaktionen
- Höhere Geschwindigkeit:
  Anwendungen laufen schneller ab, da es mehrfache/gleichzeitige Transaktionen mit einem Server ermöglicht.
- Schnellere Ausgabe:
  Durch optimierte Bildschirmausgabe und Index–Filterung ist eine schnellere Darstellung auf dem Bildschirm möglich. Außerdem vereinfacht sich die BASIC–Programmierung.
- Vermeidung von Seiteneffekten:
  Typische Seiteneffekte von Visual Basic Controls, wie Abbruch programmierter Verbindungen durch Drücken der ESCAPE–Taste, werden vermieden.

In diesem Kapitel werden zunächst die Eigenschaften des neuen Controls betrachtet, dann die zusätzlichen Events. Zum Abschluß zeigen einige Beispiele die Anwendungsmöglichkeiten des DCTL.OCX.

Eigenschaften

Die meisten Eigenschaften des DCTL.OCX Controls entsprechen denen üblicher Controls von Visual Basic. Dazu gehören

- Style properties
- Color properties
- Base properties
- Drag properties
- Font properties.

Einige Eigenschaften unterscheiden sich von denen üblicher Controls von Visual Basic:

- DDE properties
- HorAlignment property
- VertAlignment property
- WordBreak property
- TabSize property
- LastError property
- Data property
- DataToCaption property
- LinkCmd property
- LinkNext property
- LinkFilter property.

DDE properties

Zu den DDE–Properties gehören
LinkItem
LinkTopic (standardmäßig der voreingestellte NCDDE)
LinkTimeout (für die Synchronisation von LinkCmd).

HorAlignment property

Diese Eigenschaft legt die horizontale Ausrichtung der Überschrift fest:

Tabelle 8-13 horizontale Ausrichtung

Wert	Eigenschaft
LeftJustify	Linksbündig (Voreinstellung)
RightJustify	rechtsbündig
HorCenter	zentriert

VertAlignment/ Multiline property

Diese Eigenschaft legt die vertikale Ausrichtung der Überschrift fest; alternativ läßt sich auch eine mehrzeilige Anzeige einstellen. Bei mehrzeiliger Anzeige werden die Zeilenumbrüche dann von der Eigenschaft **WordBreak** bestimmt:

Tabelle 8-14 vertikale Ausrichtung

Wert	Eigenschaft
VertCenter	Vertikal zentriert (Voreinstellung)
TopJustify	oben
BottomJustify	unten
MultiLine	mehrzeilig

WordBreak property

Wenn die Eigenschaft **VertAlignment/Multiline** auf **Multiline** (Wert = 3) festgelegt ist, dann bestimmt die Eigenschaft **WordBreak** die Art des Zeilenumbruchs:

Tabelle 8-15 Art des Zeilenumbruchs

Wert	**Eigenschaft**
False	Umbruch durch CR/LF (Wagenrücklauf/neue Zeile)
True	Automatischer Umbruch, wenn das Wort nicht mehr in die Zeile paßt. Hier ist auch Umbruch durch CR/LF möglich.

TabSize Property

Diese Eigenschaft legt die Tabulator–Schrittweite fest. Standardwert sind 8 Zeichen. Der Wertebereich darf bis zu 255 gehen.

LastError property

Diese Eigenschaft ermöglicht die Ausgabe von Fehlermeldungen.
Zu Beginn einer DDE–Verbindung mit einem Server wird die Fehlervariable **LastError** zurückgesetzt (LastError = 0). Tritt während der Transaktion ein Fehler auf und wird er vom DCTL Control erkannt, ermittelt dieses einen detaillierten Fehlercode, auf den über die Eigenschaft LastError zugegriffen werden kann.

Hinweis

Das DCTL Control dekodiert keine Fehler, die als Daten übertragen werden: Dazu gehören #--Zeichen oder Leerzeichen vom NCDDE-Server.

Die Variable LastError für das DCTL–Control ist im Kapitel 11.7.1 näher beschrieben.

Data property

Die Eigenschaft **Data** dient als Argument für die folgenden DDE–Transaktionen:

Tabelle 8-16 Argumente der DDE–Transaktionen

DDE–Transaktion	Argument
Request	Angeforderter Variablenwert, wenn die Eigenschaft DataToCaption = False gesetzt ist
Advise Link	Aktualisierte Werte, wenn die Eigenschaft DataToCaption = False gesetzt ist
Poke	Zu übertragender Wert
Execute	Auszuführende Anweisung

DataToCaption property

Die Eigenschaft DataToCaption bestimmt das Ziel der Daten einer DDE–Transaktion.

Tabelle 8-17 Ziel der Daten

Wert	Bedeutung
True	Datenziel ist Caption property
False	Datenziel ist Data property

LinkCmd property

Änderungen der Eigenschaft **LinkCmd** starten die DDE–Aktivitäten des DCTL Control. Wenn keine Aktivität vorliegt, ist LinkCmd = 0.

Tabelle 8-18 Eigenschaft von LinCmd property

lfd. Nr	Wechsel zu	DDE–Aktivität	Ende durch
1	Advise Link	Advise Link ist eingerichtet. Return nach Aufbau des Advise Link. Advise Link läßt sich durch Kommando Stop löschen	Stop
2	Advise Link_ NotifyData	Wie unter Lfd.Nr. 1, zusätzlich Aktion (1), wenn DDE–Daten eintreffen	Stop
3	Advise Link_ NotifyDataWhenVisible	Wie unter Lfd.Nr. 1, zusätzlich Aktion (2), wenn DDE–Daten eintreffen	Stop
4	Advise LinkAsync	Stößt Advise Link–Einrichtung an. Return vor Aufbau des Advise Link. Advise Link läßt sich durch Kommando Stop löschen	Stop
5	Advise LinkAsync_ NotifyData	Wie unter Lfd.Nr. 4, zusätzlich Aktion (1), wenn DDE–Daten eintreffen	Stop
6	Advise LinkAsync_ NotifyDataWhenVisible	Wie unter Lfd.Nr. 4, zusätzlich Aktion (2), wenn DDE–Daten eintreffen	Stop
7	Stop	Löscht einen Advise Link. Return nach Löschen des Advise Link.	Selber
8	StopAsync	Stößt Advise Link–Löschen an. Return vor Abbau des Advise Link.	Sync
9	StopAsync_ Notify	Wie unter Lfd.Nr. 8, zusätzlich Aktion (1) nach erfolgtem Abbau des Advise Link	Sync
10	StopAsync_ NotifyWhenVisible	Wie unter Lfd.Nr. 8, zusätzlich Aktion (2) nach erfolgtem Abbau des Advise Link	Sync
11	Request	Lesen einer DDE–Variablen. Return nach erfolgtem Lesen	selber
12	RequestAsync	Lesen einer DDE–Variablen anstoßen. Return vor erfolgtem Lesen	Sync
13	RequestAsync_ Notify	Wie unter Lfd.Nr. 12, zusätzlich Aktion (1) nach erfolgtem Lesen	Sync
14	RequestAsync_ NotifyWhenVisible	Wie unter Lfd.Nr. 12, zusätzlich Aktion (2) nach erfolgtem Lesen	Sync
15	Execute	Sendet eine Anweisung an den Server. Return nach erfolgter Ausführung.	selber
16	ExecuteAsync	Veranlaßt Anweisung an den Server. Return vor erfolgter Ausführung.	Sync
17	ExecuteAsync_ Notify	Wie unter Lfd.Nr. 16, zusätzlich Aktion (1) nach erfolgter Ausführung.	Sync
18	ExecuteAsync_ NotifyWhenVisible	Wie unter Lfd.Nr. 16, zusätzlich Aktion (2) nach erfolgter Ausführung.	Sync
19	Poke	Schreiben einer DDE–Variablen. Return nach erfolgtem Schreiben. Es wird der Wert aus Data (nicht aus Caption) geschrieben.	selber
20	PokeAsync	Veranlaßt Schreiben einer DDE–Variablen. Return vor erfolgter Ausführung.	Sync

lfd. Nr	Wechsel zu	DDE–Aktivität	Ende durch
21	PokeAsync_Notify	Wie unter Lfd.Nr. 20, zusätzlich Aktion (1) nach erfolgtem Schreiben.	Sync
22	PokeAsync_ NotifyWhenVisible	Wie unter Lfd.Nr. 20, zusätzlich Aktion (2) nach erfolgtem Schreiben.	Sync
23	Sync	Schließt asynchrone Anweisungen in gleicher Weise wie synchrone ab. Keine Ausführung, wenn keine asynchrone Anweisung arbeitet.	selber

Aktionen

Die in der obigen Tabelle genannten Aktionen sind:

Aktion (1)
Es wird versucht, die Event–Prozedur **DdeNotify** aufzurufen. Wenn Visual Basic zu diesem Zeitpunkt keine Event–Prozedur aufruft oder der Parameter der Event–Prozedur nicht geändert wurde, versucht das DCTL Control zehn Mal in der Sekunde dieses Event zu schicken, solange sich der Parameter der Event–Prozedur DdeNotify nicht ändert.

Aktion (2)
Das DCTL Control ruft die Event–Prozedur **DdeNotify** auf, wenn es von WINDOWS eine Paint–Meldung bekommt. Damit sichergestellt ist, dass diese Paint–Meldungen auch erzeugt werden, ist der Bildpunkt in der oberen linken Ecke des Control ungültig, solange sich der Parameter von DdeNotify nicht ändert. Dieser Mechanismus unterdrückt auch Anzeigen, wenn das Control nicht sichtbar ist.

Hinweis

Neue DDE-Aktivitäten sollten erst begonnen werden, wenn bisherige abgeschlossen sind. Dies ist mit den in der rechten Spalte (Ende durch) genannten Parametern möglich.

Die Hotlinks aller DCTL–Controls im gleichen Fenster teilen sich eine DDE–Verbindung, wenn sie die gleiche Eigenschaft LinkTopic besitzen. Die DDE–Verbindungen der anderen Aktivitäten (außer Hotlink) werden dynamisch erzeugt und auch wieder gelöscht. Durch dieses Verhalten und die Tatsache, dass ein DCTL–Control kein Window hat, reduzieren sich die Anforderungen an die WINDOWS–Ressourcen ganz erheblich.

Hinweis

Ein Wechsel des Properties LinkCmd führt zur Auswertung der Eigenschaften LinkTopic, LinkTimeout und LinkItem. D.h. Fehler betreffend dieser Größen werden auftreten, wenn sich das Property LinkCmd ändert. Sie müssen also hier abgefangen werden.

LinkNext property

Die optional vergebbare Eigenschaft **LinkNext** hält den Namen und optional den Index eines anderen DCTL Control.
Ist LinkNext nicht leer, prüft das DCTL Control eine per Advise Link übermittelte Zeichenkette auf Indexspezifikationen des NCDDE (5 Ziffern, abgeschlossen mit Doppelpunkt). Es teilt den Text auf in indizierte Textteile und gibt diese weiter in eine verkettete Liste von DCTL Controls. Das Control, dessen Eigenschaft **LinkFilter** diesem Index entspricht, bekommt den Text. Nicht auf diese Weise zugeteilte Texte gehen verloren.

LinkFilter property

Der Wert **LinkFilter** kann zwischen 0 und 65535 liegen. Seine Verwendung ist im vorigen Abschnitt LinkNext gezeigt.

Events für DCTL.OCX

Die meisten Events des DCTL–Control sind identisch mit denen anderer Standard–Controls von Visual Basic, wie

- Click
- DblClick
- MouseDown
- MouseMove
- MouseUp
- DragDrop
- DragOver
- KeyDown
- KeyPress
- KeyUp.

Event DdeNotify

Speziell für die DDE–Kommunikation ist das Event **DdeNotify** realisiert worden: Es zeigt die Ankunft neuer Advise Link–Daten oder den Abschluß einer asynchronen DDE–Transaktion an. Seine Anwendung ist unter der Eigenschaft LinkCmd (Aktionen (1) und (2)) beschrieben.

Syntax

Sub *ctlname*_DdeNotify (*Flag* As Integer)
mit dem Argument Flag, das dem DCTL–Control anzeigt, wenn das Event wirklich auf der Basic–Ebene angekommen ist.
Es wird erwartet, dass der Wert von Flag bei jedem Aufruf der Event–Prozedur geändert wird. Denn das DCTL–Control löst das Ereignis DdeNotify solange aus, bis sich das Flag–Argument ändert. Findet diese Änderung nicht statt, ergibt sich eine kooperative Daueraktivität, die das System belastet.

8.9.4 Anwendungen von DCTL.OCX

Lesen und Anzeigen einer Variable

Es soll eine DDE–Variable sofort gelesen und am Bildschirm dargestellt werden: Dann ist dazu ein DCTL–Control (z.B. unter dem Namen DCTL1) an geeigneter Stelle auf dem Bildschirm zu plazieren. Der zugehörige Code sieht dann folgendermaßen aus:

Beispiel 8–43 Lesen und Anzeigen einer Variable

```
Sub Form_Load ( )
        Dctl1.LinkItem = ”/Channel/Parameter/R[1]” ’ the variable name
        Dctl1.DataToCaption = TRUE ’that’s default, can be omitted
        Dctl1.LinkCmd = 11 ’ commands the reading
        ’ here Dctl1.Caption holds the value of the DDE variable
End Sub
```

Lesen einer Variable ins Data Property

Es soll eine DDE–Variable sofort gelesen und verarbeitet werden, ohne sie am Bildschirm darzustellen. Dann ist dazu ein DCTL–Control (z.B. unter dem Namen DCTL2) vom Typ LABEL in einer Form anzuordnen. Der zugehörige Code sieht dann folgendermaßen aus:

Beispiel 8–44 Lesen ins Data Property

```
Sub Form_Load ( )
        Dctl1.LinkItem = ”/Channel/Parameter/R[1]” ’ the variable name
        Dctl1.DataToCaption = FALSE ’ routing data to the Data property
        Dctl1.LinkCmd = 11 ’ commands the reading
        ’ here Dctl1.Data holds the value of the DDE variable
End Sub
```

DDE–Variable schreiben

Es ist eine DDE–Variable zu schreiben. Dann ist dazu ein DCTL–Control (z.B. unter dem Namen DCTL3) vom Typ LABEL in einer Form anzuordnen. Der zugehörige Code sieht dann folgendermaßen aus:

Beispiel 8–45 Schreiben einer Variable

```
Sub Form_Load ( )
        Dctl1.LinkItem = ”/Channel/Parameter/R[1]” ’ the variable name
        Dctl1.Data = 12 ’ the value
        Dctl1.LinkCmd = 19 ’ commands the writing
        ’ here the NC variable is already successfully set to 12
End Sub
```

Kommando ausführen

Es ist ein DDE–Kommando an einen Server zu übermitteln. Dann ist dazu ein DCTL–Control (z.B. unter dem Namen DCTL4) vom Typ LABEL in einer Form anzuordnen. Der zugehörige Code sieht dann folgendermaßen aus:

Beispiel 8–46 Kommando ausführen

```
Sub Form_Load ()
        Dctl1.Data = ”Pi_start(/NC,001,_N_SET_OF)” ’ the command
        Dctl1.LinkCmd = 15 ’ sends the command
        ’ here the command is already successfully executed
End Sub
```

DDE–Hotlink anzeige

Es ist der Wert einer DDE–Variablen auf dem Bildschirm darzustellen. Dann ist dazu ein DCTL–Control (z.B. unter dem Namen DCTL4) an geeigneter Stelle auf dem Bildschirm zu plazieren. Es ist ein Hotlink einzurichten, der dann als Hintergrundaktivität des DCTL–Controls läuft. Der zugehörige Code sieht dann wie im folgenden Beispiel aus.
Es ist aber auch möglich, die codierte Property–Einstellung zur Entwurfszeit auszuführen.

Beispiel 8–47 Hotlink mit DCTL

```
Sub Form_Load ( )
Dctl1.LinkItem = ”/Channel/Parameter/R[1]” ’ the variable name
Dctl1.DataToCaption = TRUE ’ that’s default, can be omitted
Dctl1.LinkCmd = 4 ’ initiates the creation of a hotlink
End Sub
```

Beschleunigung durch Parallelarbeit

Beim Laden einer Form sind einige unabhängige DDE–Aktivitäten auszuführen. In diesem Fall ist ein schnelles Laden der Form wünschenswert. Dies wird – im Zusammenspiel mit dem DCTL–Control – am besten durch parallel laufende DDE–Aktivitäten erreicht. Im nächsten Beispiel ist ein geeigneter Code dargestellt.

Beispiel 8–48 Beschleunigung durch Parallelarbeit

```
Sub Form_Load ( )
’ start reading variable 1
Dctl1.LinkItem = ”/Channel/Parameter/R[1]” ’ the variable name
Dctl1.LinkCmd = 12
’ initiates the reading
’ start reading variable 2
Dctl2.LinkItem = ”/Channel/Parameter/R[2]” ’ the variable name
Dctl2.LinkCmd = 12
’ initiates the reading
’ start reading variable 3
Dctl3.LinkItem = ”/Channel/Parameter/R[3]” ’ the variable name
Dctl3.LinkCmd = 12
’ initiates the reading
’ start a hotlink into display
Dctl4.LinkItem = ”/Channel/Parameter/R[4]” ’ the variable name
Dctl4.DataToCaption = TRUE
’ that’s default, can be omitted
Dctl4.LinkCmd = 4
’ creates the hotlink
’ start a command execution
Dctl5.Data = ”Pi_start(/NC,001,_N_SET_OF)” ’ the command
Dctl5.LinkCmd = 16
’ commands execution
’ here the tree variable accesses, the hotlink creation and the ’ ’command are
working in parallel. You can not be sure that any of ’ ’ them has completed.
Dctl1.LinkCmd = 23
’ wait until variable 1 read
Dctl2.LinkCmd = 23
’ wait until variable 2 read
Dctl3.LinkCmd = 23
’ wait until variable 3 read
Dctl5.LinkCmd = 23
’ wait until command executed
’ here the variable accesses and the command have completed, the hotlink will
’ show it’s value on screen as soon as possible.
End Sub
```

Beschleunigung durch Textanordnung

Es sind viele Daten mit hoher Frequenz zu lesen und anzuzeigen. Für eine Anzeige sollte die Sprache BASIC nicht verwendet werden. Außerdem sollte der Umfang der übertragenen Daten minimiert werden. Auf der NCDDE–Seite sorgen Feldzugriff auf Daten und eine Kombination von Feldzugriff mit Datenvorbereitung für die Erfüllung dieser Forderungen. Das DCTL–Control bietet dazu mehrzeilige Darstellung und Index–Filterfunktionen.

Beispiel 8–49 Beschleunigung durch Textanordnung

```
'NCDDE array access with "Field" data preparation - Dctl index filtering:
' high frequency display of 5 values in 5 different controls
Dctl1.LinkItem = "/Channel/Parameter/R[1,5](!""!d%12.5g"")" ' variable
Dctl1.LinkFilter = 1           '     index of accepted data
Dctl1.LinkNext = "Dctl2"       '     linkage to the next control
Dctl2.LinkFilter = 2           '     index of accepted data
Dctl2.LinkNext = "Dctl3"       '     linkage to the next control
Dctl3.LinkFilter = 3           '     index of accepted data
Dctl3.LinkNext = "Dctl4"       '     linkage to the next control
Dctl4.LinkFilter = 4           '     index of accepted data
Dctl4.LinkNext = "Dctl5"       '     linkage to the next control
Dctl5.LinkFilter = 5           '     index of accepted data
Dctl1.LinkCmd = 4              '     initiates the creation of a hotlink
NCDDE array access - Dctl multiline display:
' high frequency display of 5 values in a column
Dctl1.LinkItem = "/Channel/Parameter/R[1,5]("" !d%12.5g"
Dctl1.LinkItem = Dctl1.LinkItem + Chr$(13) + Chr$(10)+""")"
Dctl1.DataToCaption = TRUE ' that's default, can be omitted
Dctl1.VertAlignment = 3 ' multiline selection
Dctl1.LinkCmd = 4 ' initiates the creation of a hotlink
```

Benachrichtigung bei Änderung

Wenn das Layout einer Bildschirmanzeige von einer Variablen abhängt, auf die über DDE zugegriffen wird, dann wird man sich diese Variable über Hotlink im DCTL–Control holenen. Mit der Benachrichtigung einer Variablenänderung läßt sich dann der Bildschirminhalt neu arrangieren. Da dieser Vorgang recht zeitaufwendig abläuft, sollte er nur ausgeführt werden, wenn die Form sichtbar ist.

Beispiel 8–50 Benachrichtigung bei Änderung

```
Sub Form_Load ( )
‘basic code that creates a hotlink with notification “when visible”
Dctl1.LinkItem = “/Channel/Parameter/R[1]” ‘the variable name
Dctl1.LinkCmd = 6 ‘initiates the creation of a hotlink
‘handler for the notification event
End Sub
Sub Dct1_DdeNotify ( Index As Integer, Flag As Integer )
Flag = Flag + 1 ‘Flag MUST change
... ‘rearrangement to be done
End Sub
```

Fehlerbehandlung

Hier wird die typische Fehlerbehandlung für Lesen, Schreiben und Ausführen gezeigt

Beispiel 8–51 Fehlerbehandlung

```
On Error Goto TypicalErrorHandling
Dctl1.LinkCmd = 11 ’ a DDE activity
...
TypicalErrorHandling:
Select Case Dctl1.Lasterror \ 16777216’ selection by error source
        Case 2 ’ MPI level error
        ... ’ e. g. no connection to NC
        Case 3 , 5 ’ NC/PLC level error
        ... ’ e. g. non existing variable
        Case 7 ’ Dctl level error
        Select Case Dctl1.Lasterror MOD 256 ’ selection by error code
        Case 7 ’ Dctl level timeout occurred
        ...
        Case Else ’ other Dctl level errors
        ...
        End Select
        Case Else ’ other error sources
        ...
End Select
...
```

8.10 Diagnose - Möglichkeiten für NCDDE–Zugriff

8.10.1 Testfunktionen des NCDDE–Servers

Übersicht

Die Testfunktionen des NCDDE–Servers geben unter anderem Auskunft über die lokalen und externen Variablen, die zum Zeitpunkt des Entstehens der Datei im NCDDE–Server bekannt waren. Diese Testfunktionen werden folgendermaßen aufgerufen:

1. NCDDE–Server in der Programmgruppe SINUMERIK 840D MMC–OEM starten
2. Mit ALT+TAB zum Programm NCDDE - NC Communication DDE–Server schalten, Ikone entsteht
3. Klick auf Ikone: Fenster mit folgendem Inhalt erscheint.

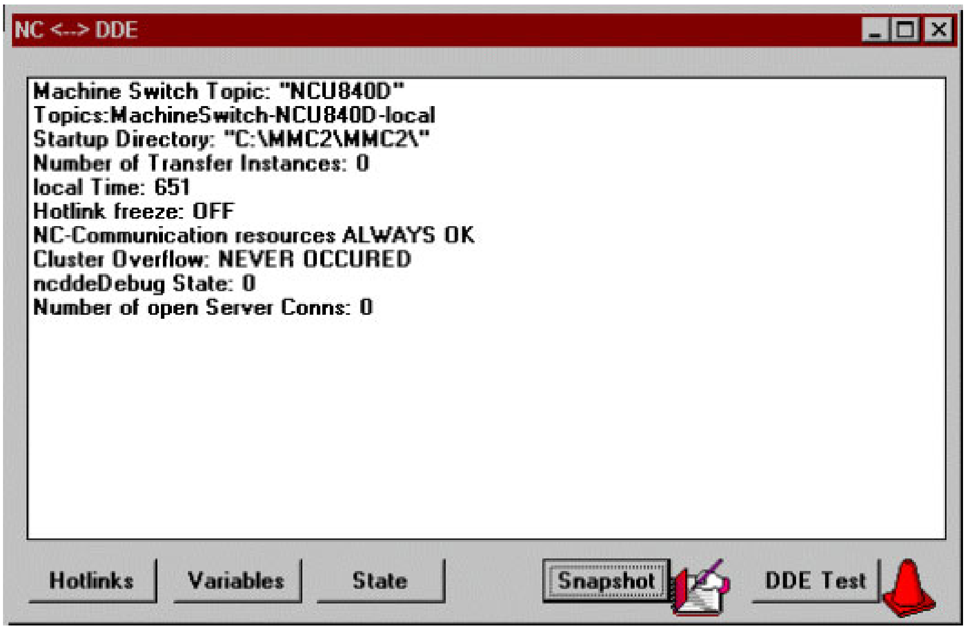


Bild 8–2 NCDDE–Server Standardanzeige

Folgende Funktionen sind vorzugsweise für eine Fehlersuche in der Umgebung des NCDDE–Servers vorgesehen.

Hotlinks

Dies ist eine Liste, in der Advise–Links (Hotlinks und Warmlinks) zusammengefaßt sind. Sie ist als Tabelle mit fünf Spalten strukturiert, die folgende Bedeutung haben:

Tabelle 8-19 Hotlinks

Spalte	Info	Bemerkungen
1	PDU–Referenz	interner Wert: Mögliche PDU–Referenz zur Kommunikation mit NCK und PLC
2	Advise Link	LOCAL Link auf lokale Variable REMOTE Link auf externe Variable PILED Doppelter Externer Advise–Link, wurde auf einen anderen Auftrag aufgeschaltet.
3	Update–Zeit	Zeitpunkt der letzten Refresh–PDU in einem internen Zeitmaß des NCDDE–Servers
4	Lasterror–Variable	Lasterror–Angabe gemäß Kapitel 11.7 dieser Doku. Sie deckt sich nicht immer mit dem an der DDE–Schnittstelle des Servers abfragbaren Wert, da an dieser der letzte Fehler mehrerer Transaktionen zu einer Verbindung abfragbar ist.
5	Variablen–Name	Variablenbezeichnung gemäß Kapitel 11.

Variables

Zeigt die Variablen an, zu denen der NCDDE–Servers Verbindung hält und wo diese liegen: “LOCAL“ oder “PLC/NC“

Snapshot

Beim Drücken dieses “Buttons“ wird eine Datei mit dem Namen “NCDDE_X.TXT“ generiert, welches den Status, Hotlinks und die Variablen des NCDDE–Servers enthält.

DDE Test

Startet ein Testprogramm “DDETEST.EXE“ mit folgenden Funktionen

Tabelle 8-20 DDE–Test Kommandos

Command	Aktion	Bedeutung
Passive	keine	Grundstellung, keine Funktion aktiv
Hotlink	Start	Advise Link aufbauen
Request	DoIt	Variable lesen
Poke	DoIt	Variable schreiben
Execute	DoIt	Ausführung eines Dienstes

Unter **Service|Topic** ist die angeschlossene NC zu kennzeichnen, z.B. **NCDDE|NCU840D**. “DEFAULT_NC“ liest die Einstellungen aus der Datei “MMC.INI“
Die Funktion **Command** wird durch Anklicken zwischen den 5 Möglichkeiten umgeschaltet.
Die Fehlermeldungen **LastError** werden im Kapitel 11.7 beschrieben.

8.10.2 Verbindungszustand

Variable NcState

Der Server veröffentlicht den Verbindungszustand zur CNC über die serverlokale Variable NcState. Diese Variable existiert sofort nach dem Start des Servers. Sie unterscheidet sich von anderen serverlokalen Variablen nur dadurch, dass sie über die DDE–Schnittstelle nicht verändert werden kann.
Die Variable kann folgende Werte annehmen :

Tabelle 8-21 Werte der Variablen NcState

Wert	Bedeutung
0	Normalbetrieb
1	einige Verbindungen zu CNC gestört
2	alle Verbindungen zu CNC gestört
3	Interpretation der Startup Datei
4	Server Initialisierung

8.10.3 Behandlung von Störfällen

Fehler–Quittungen von NCK

Fehlerbedingungen wie Ressourcen–Mangel, Zugriffsschutz–Verletzung, falscher Betriebszustand, usw. werden vom NCK über Quittungen mitgeteilt. Kann der NCDDE–Server diese Fehlerbedingungen nicht bearbeiten, dann werden die entsprechenden Transaktionen Request, Peek und Execute der DDE–Schnittstelle mit Fehler beendet, d.h. man bekommt kein Ergebnis.

Variable LastError

Eine detaillierte Diagnose liefert die Variable LastError, die zu der letzten Transaktion einer Verbindung existiert. Sie kann über das LinkItem LastError ausgelesen werden. Nach dem Auslesen wird die Variable auf Null gesetzt. Sie zeigt immer den letzten im NCDDE–Server registrierten Fehler an.
Die Variable LastError besteht aus vier Byte, wobei pro Byte in absteigender Reihenfolge (High Byte è Low Byte) die folgenden Fehlergruppen zu finden sind:

- Übergeordnete Fehlerklasse, Fehlerquelle
- Fehler–Region
- Fehlerklasse
- Fehlercode

Die Bedeutung der einzelnen Fehlercodes ist im Kapitel 11.7 unter NCDDE–Fehlermeldungen zu finden.

Verbindungsabbruch zum NCK

Beim Unterbrechen der Verbindung quittiert der NCDDE–Server laufende Transaktionen Request, Poke und Execute negativ. Während eine Verbindung abgebrochen ist, lehnt er die Ausführung von Transaktionen ab, gleichzeitig versucht er immer wieder die Verbindungen zum NCK wieder aufzubauen. Der Verbindungszustand wird in der serverlokalen Variablen NcState veröffentlicht.

Behandlung von Advise Links

Ist bei Advise Links die Verbindung zum NCK unterbrochen, entspricht der vom NCDDE–Server gelieferte Wert dem Zeichen ‘#’. Advise Links werden nach Verbindungswiederaufbau auf der CNC wieder eingerichtet.

Ressourcen–Mangel im NCDDE–Server

Tritt ein Ressourcen Mangel im NCDDE–Server auf, dann werden die entsprechenden Transaktionen an der DDE–Schnittstelle mit Fehlerkennung beendet.

8.11 NCDDE–Server Konfiguration für Netzwerkzugriffe

...
nur für Entwicklungszwecke

8.12 Erweiterungen des NCDDE–Servers

8.12.1 Multivariablendienst

Übersicht

Der Multivariablendienst erlaubt den DDE–Zugriff auf mehrere Variablen mit einem NCDDE–Auftrag. Er beschleunigt den Zugriff auf mehrere Einzelvariablen, ist allerdings nur für Lese– und Schreiboperationen einsetzbar (nicht Hotlink).
Als Item sind die mit '|' getrennten Items der entsprechenden Einzelvariablen/Array Zugriffe anzugeben. Die beim Lesezugriff gelieferten Daten sind dicht gepackt. Trennzeichen müssen wie bisher bei Arrayzugriffen über Formatangaben oder über eine neue Zugriffsmodifikation (siehe Kapitel 8.12.3) parametriert werden. Bei Schreibzugriffen wird das erste Zeichen der gelieferten Daten als Trennzeichen für die einzelnen Datensätze interpretiert.

Grenzen

- Je Auftrag werden **max. 8** dicht gepackte PDUs abgesetzt. Damit können normalerweise mehr als 100 Einzelzugriffe in einem Auftrag abgewickelt werden. (Die genaue Anzahl ist durch Versuch zu ermitteln)
- Die PDUs werden alle an **eine Zieladresse** gesandt. Deshalb können PLC– und NC–Zugriffe nicht in einem Auftrag gemischt werden. Darüber hinaus dürfen kanalspezifische Zugriffe für unterschiedliche **Kanäle nicht gemischt** sein. (NC–Anforderung). Gleiches gilt für Zugriffe auf antriebsspezifische Variablen.
- Es können **nur echte Variablen** (BTSS–Schnittstelle/PLC–BUB) mit dem Multivariablendienst adressiert werden. Ein Zugriff auf Datum u. Uhrzeit, Systemzustandsliste, Directory–Information, ... ist damit nicht möglich.
- Achtung, die DDE **Itemgröße ist auf 255 Zeichen begrenzt**. Überschreitet der Itemstring diese Größe, dann muss das Item indirekt (siehe Kapitel 8.12.2) angegeben werden.

Beispiel für Schreiben und Lesen mit dem Multivariablendienst
Item:
/channel/parameter/r[1,2](|"!l%ld")|/channel/parameter/r[10](|)
Daten z.B.: |1|2|10.000000

8.12.2 Indirekte Item–Ausgabe

Die indirekte Item–Angabe erlaubt Items von mehr als 255 Zeichen (bis 4KB). Der Inhalt einer NCDDE–lokalen Variablen kann als Item für DDE–Zugriffe verwendet werden. In diesem Fall muss der Name der lokalen Variablen mit vorangestelltem ’>’-Zeichen als Item spezifiziert werden.

Beispiel für den Zugriff auf R10:

Exec:	NEW(x,“/channel/parameter/r[10]“)
Item:	>x
Daten z.B.	10.000000

Hinweis

Die Datenlänge ist beim Schreiben von Variablen und beim Ausführen von Kommandos
im NCDDE-Server auf 4KB begrenzt. Beim Überschreiten dieses Wertes kommt die Fehlermeldung 0X01050414.

8.12.3 Neue Zugriffsmodifikationen

Durch die Steuerzeichen **’|’** und **’^’**, in runden Klammern an den Itemstring angefügt, läßt sich der Zugriff folgendermaßen modifizieren:

’|’ Fügt bei CF_TEXT–Lesezugriffen ein ’|’ Zeichen vor jedem Einzeldatum ein. Dies wird bei Schreibzugriffen nicht ausgewertet (siehe das Beispiel im Multivariablendienst Kapitel 8.12.1).

’^’ Für eine derartig gekennzeichnete Variable ist die Hotlink–Abschaltung (DEBA/DEBR) unwirksam.

8.13 Zugriff auf globale Anwendervariablen GUD, SGUD, MGUD, UGUD, GD3 bis GD9

Übersicht

Globale Daten gibt es für den NCK und jeweils für einen Kanal.
Die NCK–spezifischen globalen Anwendervariablen sind einmal in der Steuerung vorhanden. Sie eignen sich für kanalunabhängige Einstellungen oder zur Programmkoordinierung.

Kanalspezifische globale Anwendervariablen sind einmal pro Kanal vorhanden. Sie eignen sich für kanalspezifische Einstellungen oder zur Datenübergabe zwischen den Programmen, die in einem Kanal ablaufen.
Das gleiche Verfahren läßt sich auch auf lokale Anwenderdaten anwenden. Hier gelten sinngemäß die gleichen Aussagen.
Damit der NCDDE–Server auf Anwendervariable zugreifen kann, sind sie zunächst zu definieren und dann zu aktivieren. Für die Clusterung von Variablen sind dann noch NSK–Dateien zu erzeugen und einzubinden. Dies erfolgt in fünf Schritten:

1. Erstellen einer Definitionsdatei
2. Kopieren dieser Definitionsdatei in das Verzeichnis /_N_DEF_DIR des NCK
3. Aktivieren der Anwenderdaten als *.ACC–Datei mit INITIAL.INI
4. Erzeugen der *.NSK–Datei mit dem MAP–Kommando
5. Aufnahme des erzeugten *.NSK–Datei in die NSK–Datei des NCDDE–Servers.

Definitionsdatei

Die Definition von globalen Anwendervariablen erfolgt in Definitionsdateien (Bausteinen) mit festgelegten Namen:

- _N_GUD_DEF für GUD
- _N_SGUD_DEF für GD1 = SGUD globale Daten Siemens
- _N_MGUD_DEF für GD2 = MGUD globale Daten Maschinenhersteller
- _N_UGUD_DEF für GD3 = UGUD globale Daten Anwender
- _N_GUD4_DEF bis _N_GUD9_DEF für GD4 bis GD9

Diese Bausteine liegen im festen Verzeichnis /_N_DEF_DIR des NCK.
Die Gesamtzahl der Dateien für globale Daten hängt vom allgemeinen Maschinendatum 18118 (MM_NUM_GUD_MODULES) ab (näheres siehe Inbetriebnahmeanleitung). Voreingestellt ist der Wert 4.

Globale Daten definieren

Globale Daten werden definiert über:

• Definitionskopf	DEF
• Bereich	NCK oder CHAN
• Zahlentyp	z.B. REAL oder INT
• Variablenname	z.B. RUECKZUG
• Parameter	in eckigen Klammern
• Kommentar	Text mit Semikolon am Anfang

Weitere Erläuterungen sind der Programmieranleitung zu entnehmen.

Definitionsdateien erstellen

Definitionsdateien lassen sich im NCK oder im MMC erstellen.
Im NCK:
Eine Definitionsdatei für globale Variable, wie sie auf Teileprogramm–Ebene im NCK erzeugt werden kann, gehört in das Verzeichnis /_N_DEF_DIR und besteht aus:

- Programm–Kennzeichnung in der ersten Zeile
- Kommentarzeile mit Pfadangabe (wird ausgewertet)
- Definitionen
- Abschluß durch M02, M17 oder M30.

Beispiel 8–55 Definition globaler Variable im NCK

```
%_N_MGUD_DEF
; $PATH=/_N_DEF_DIR
DEF NCK REAL RUECKZUG ; Definition globaler Variable für NCK
DEF CHAN INT TABELLE[100] ; Definition kanalspezifischer Variable
DEF CHAN REAL BLF_OFFS_X
M17 ; Abschluß dieser Zeile mit RETURN nötig
```

Im HMI:
Eine Definitionsdatei für globale Variable mit dem Dateinamen MGUD.DEF, wie sie im HMI erzeugt werden kann, liegt z.B. im Verzeichnis C:\TMP und besteht aus:

- Definitionen
- Abschluß durch M02, M17 oder M30.

Beispiel 8–56 Definition globaler Variable im HMI

```
DEF NCK REAL RUECKZUG ; Definition globaler Variable für NCK
DEF CHAN INT TABELLE[100] ; Definition kanalspezifischer Variable
DEF CHAN REAL BLF_OFFS_X
M17 ;Abschluß dieser Zeile mit RETURN nötig
```

Hinweis

Diese Datei muss vom MMC noch in das Verzeichnis /_N_DEF_DIR des NCK transferiert werden. Dies geht mit dem Domain-Dienst COPY_TO_NC:COPY_TO_NC(C:\TMP\MGUD.DEF,/NC/_N_DEF_DIR/Ä_N_MGUD_DEF,trans)

Anwenderdaten aktivieren

Die Anwenderdaten werden durch Kopieren einer Datei mit dem Namen INITIAL.INI in den NCK aktiviert. Diese Datei kann sehr kurz sein: Ein Eintrag von M17, gefolgt von RETURN, reicht völlig aus. Für eine Datei INITIAL.INI, die im Verzeichnis C:\TMP steht, gilt:

```
COPY_TO_NC(C:\TMP\INITIAL.INI, /NC/_N_INITIAL_INI, Ätrans)
```

Damit werden zwei ACC–Dateien erzeugt mit den Namen:
_N_NC_GD2_ACC für die globalen Anwendervariablen
_N_CH_GD2_ACC für die kanalspezifischen Anwendervariablen
(gilt für das obige Beispiel mit MGUD = GD2).

Hinweis

Vor dem Kopieren der Datei INITIAL.INI alle Programme, Frames und Maschinendaten sichern, da der statische Speicher formatiert wird.

NSK–Datei für NCK erzeugen

Durch Aufruf des MAP–Kommandos werden aus den ACC–Dateien gleichnamige NSK–Dateien für globale Anwendervariable des NCK erzeugt.
Das Beispiel zeigt den Aufruf unter Visual Basic.

Aufrufen des “MAP_ACC_NC“–Kommandos

C:\MMC2\MGUD_NCK.NSK :	Dateiname in der WINDOWS–Umgebung
/NC/_N_NC_GD2_ACC	: NC–Domain
trans	: Variable TransferState
0	: Bereich NCK
2D	: Bausteintyp MGUD
10	: Zeitüberwachung der Transaktion mit 10s
/ACC/NCK/MGUD/ :	beliebige, vom Anwender wählbare Zeichenkette, die vor die Anwendervariable gestellt wird.

Beispiel 8–57 Erzeugung einer NSK–Datei für NCK

```
Sub Form_Load ( )
        Label1.LinkTopic = ”NCDDE|MMC2HW0”
        Label1.LinkMode = 2
        Label1.LinkExecute ”MAP_ACC_NC(C:\MMC2\MGUD_NCK.NSK,
                Ä/NC/_N_NC_GD2_ACC, trans, 0, 2D , 10,
                Ä/ACC/NCK/MGUD/)”
End Sub
```

NSK–Datei für Kanal erzeugen

Durch Aufruf des MAP–Kommandos werden aus den ACC–Dateien gleichnamige NSK–Dateien für globale Anwendervariable des Kanals erzeugt. Das Beispiel zeigt den Aufruf unter Visual Basic.

Aufrufen des “MAP–Kommandos“

C:\MMC2\MGUD_CH.NSK	: Dateiname in der WINDOWS–Umgebung
/NC/_N_CH_GD2_ACC	: NC–Domain
trans	: Variable TransferState
2	: Bereich Kanal
2D	: Bausteintyp MGUD
10	: Zeitüberwachung der Transaktion mit 10s
/ACC/CH/MGUD/	: beliebige, vom Anwender wählbare Zeichenkette, die vor die Anwendervaria–ble gestellt wird.

Beispiel 8–58 Erzeugung einer NSK–Datei für Kanal

```
Sub Form_Load ()
        Label1.LinkTopic = ”NCDDE|MMC2HW0”
        Label1.LinkMode = 2
        Label1.LinkExecute
”MAP_ACC_NC(C:\MMC2\MGUD_CH.NSK,/NC/_N_CH_GD2_ACC
        ,trans,2,2D,10,/ACC/CH/MGUD/)”
End Sub
```

Hinweis

Die NSK-Datei wird im Binärformat (*.MAP) und im ASCII-Format (*.NSK) erzeugt.

Aufnahme in NSK–Datei des NCDDE–Servers

Die in diesem Beispiel entstandenen Dateien MGUD_NCK.NSK und MGUD_CH.NSK sind in die NSK–Datei des NCDDE–Servers NCDDE311.NSK einzufügen mit:

```
REM IMPORT ADDITIONAL USER VARIABLES
CALL(MGUD_NCK.NSK)
CALL(MGUD_CH.NSK)
REM
```

Zugriff auf NCK–Anwendervariable

Am Beispiel der Variablen RUECKZUG soll gezeigt werden, wie eine NCK–Anwendervariable ausgelesen wird.

Beispiel 8–59 Zugriff auf NCK–Anwendervariable RUECKZUG

```
Sub Form_Load ( )
        CtlName1.LinkTopic = g_chNCDDEServiceName
        CtlName1.LinkItem = ”/acc/nck/mgud/RUECKZUG”
        CtlName1.LinkMode = 2
        CtlName1.LinkRequest
        CtlName1.LinkMode = 0
End Sub
```

Zugriff auf Kanal–Anwendervariable

Am Beispiel der Variablen BLF_OFFS_X soll gezeigt werden, wie eine Kanal–Anwendervariable ausgelesen wird.

Beispiel 8–60 Zugriff auf Kanal–Anwendervariable BLF_OFFS_X

```
Sub Form_Load( )
        CtlName.LinkTopic = g_chNCDDEServiceName
        CtlName.LinkItem = ”/acc/ch/mgud/BLF_OFFS_X[u2]” ’für 2.Kanal
        CtlName.LinkMode = 2
        CtlName.LinkRequest
        CtlName.LinkMode = 0
End Sub
```

Hinweis

Weitere Informationen zum Anlegen und der Anwendung von Anwenderdaten sind in der Inbetriebnahme-Anleitung /IAD/ und in der Programmieranleitung /PA/ zu finden.

8.14 Variablen–Online–Hilfe

Übersicht

Die Variablen–Online–Hilfe unterstützt den OEM–Programmierer bei der Auswahl und Definition von Daten aus dem NCK–Bereich. Sie ist aufgebaut wie alle anderen Hilfe–Dateien unter WINDOWS und bietet den gleichen Funktionsumfang. Die Variablen–Online–Hilfe ist unabhängig vom OEM–Paket HMI und als Hilfe–Datei unter dem Namen BTSS_GR.HLP (mit deutschen Texten) im Verzeichnis HLP gespeichert.

Zielsysteme

Der Einsatzbereich der Variablen–Online–Hilfe ist nicht auf die OEM–Programmierung der Anwenderoberfläche des HMI beschränkt: Sie ist auch geeignet für die Projektierung des MMC 100 und des NC–Var–Selektors in der PLC–Programmierumgebung.

Funktionsumfang

Die Variablen–Online–Hilfe bietet Informationen zu allen NCK–Variablen, wie sie im Kapitel 11 zusammengestellt und im Listenbuch /LIS/ näher erläutert sind.
Der Anwender kann über mehrere Beschreibungsebenen zu den Informationen über eine Variable gehen.
Ausgehend vom Datenbereich mit:
Datenbereich Baustein Variable Beispiel
oder alphabetisch sortiert über Baustein:
Baustein Variable Beispiel
oder mit der Funktion **SUCHEN (FIND)** nach Schlüsselworten.
Schlüsselworte sind:

Kurzbeschreibung der Variablen	z.B. **Spindeltyp**
Variablenname	z.B. **Variable spindleType**
Kurzbezeichnung des Bausteins	z.B. **SSP** (für Spindelzustandsdaten).

Daten übernehmen

Aus dem angezeigten Hilfethema lassen sich Teile kopieren und in andere Dateien einfügen. Dies eignet sich besonders dafür, die Beispiele der Variablen–Online–Hilfe direkt in eigene OEM–Programme zu übernehmen. Dazu ist folgendermaßen vorzugehen:
Menü **Bearbeiten** anwählen
Menüpunkt **Kopieren** anwählen
Benötigten Text mit Maus auswählen
Schaltfläche **Kopieren** betätigen
zur anderen Applikation wechseln
Text einfügen.

Weitere Funktionen

Mit der Variablen–Online–Hilfe können weiterhin

- Themen gedruckt
- Anmerkungen zu allen Themen eingefügt
- Lesezeichen zum schnellen Finden oft benötigter Infos definiert werden.

Hinweis

Die Anmerkungen zur Variablen-Online-Hilfe werden in der Datei BTSS_VAR.ANN (ANN ist Abkürzung von Annex) und die Lesezeichen in der Datei WINHELP.BMK (BMK ist Abkürzung von Bookmark) im WINDOWS-Verzeichnis gespeichert.

8.15 Troubleshooting

8.15.1 keine Verbindung zu NCK/PLC

- Verbindungskabel
- MPI–Treiber müssen installiert sein
- MMC.INI Prüfen
- WINSTART.BAT
- S7DPMPI.INI

8.15.2 ...didn't respond to DDE–Initiate

- Link–Topic prüfen
- Link–Item prüfen
- ist die Variable angelegt, speziell bei PLC–Zugriffen: Ist der Datenbaustein angelegt

8.15.3 Form Load dauert lange durch Aufbauen vieler Hotlinks

- DCTL–Control verwenden
- asynchrone Hotlinks einrichten

8.15.4 Execute–Kommando funktioniert bei ersten Mal nicht

Ursache

Bei manchen Kommandos erwartet der NCDDE–Server, dass bereits eine Verbindung zur NC besteht.

Lösung

Zuvor einen Hotlink auf eine NC –Variable anlegen.

8.16 Ermittlung der aktiven Busteilnehmer

In der Systemsteuerung den Ordner „PG/PC–Schnittstellen einstellen“ öffnen. Im daraufhin aufgeblendeten Dialogfenster in der Liste „Benutzte Schnittstellenparametrierung“ die aktive Kommunikationsschnittstelle (z.B. CP5611 (MPI) <Aktiv>) selektieren und die Schaltfläche „Diagnose“ drücken. Mit der Schaltfläche „Testen“ im nachfolgend angezeigten Dialog kann die Funktionsfähigkeit der selektierten Kommunikationsschnittstelle festgestellt werden. Mit der Schaltfläche „Lesen“ können die aktiven Busteilnehmer ermittelt werden.
Programmtechnisch können die aktiven Busteilnehmer durch Auslesen der Variablen /Nck/Nck/BusState ermittel werden. Geliefert wird ein 32–Nit Wert, bei dem gesetzte Bits einen aktiven Busteilnehmer anzeigen (Bit0 gesetzt = aktiver Teilnehmer auf Busadresse 0, Bit1 gesetzt = aktiver Teilnehmer auf Busadresse 1, ...). Die Funktionalität steht am MPI– und Profibus zur Verfügung. Es werden die 32 „unteren“ Adressen des lokalen Bussegments betrachtet.

A.3 Abkürzungen

Tabelle A-1 Abkürzungen

Abkürzung	**Bedeutung**
DockPos	Docking Position, Haltestelle
FLR	Fertigungsleitrechner
FTP	File Transfer Protocol
MMC	Men-Machine-Communication, Mensch-Maschine-Kommunikation
NCU	Numerical Control Unit, Kontrolleinheit für die Achs-/Spindelbewegungen
PLC	Programmable Logic Control, Maschinensteuereinheit
RKS	Rechnerkopplungssoftware in MMC102, MMC103
RPC	Remote Procedure Call, Funktionsaufruf über Netzverbindung
TCP/IP	Transmission Control Protocol / Internet Protocol
TPS	Transportsystem
WPC	Workpiece Carrier, Werkstückträger
WZ	Werkzeug

A.4 Fehlernummern

Tabelle A-2 RPC SINUMERIK Fehlernummern

Fehler- nummer	Bedeutung
-70	ERR_WAIT_TIMEOUT Kommt im Logfile nach Ablauf einer Wartezeitnach einem erfolglosen Sendeversuch
-97	ERR_RESTARTING RPC-Returnwert, wenn während des Starts bereits ein RPC kommt
-98	ERR_ORDERLIST_FULL RPC-Returnwert, wenn die RPC SINUMERIK interne Auftragsliste voll ist, → warten und dann den Call wiederholen
-99	ERR_NOT_SUPPORTED RPC-Returnwert, wenn RPC nicht unterstützt wird. z.B. T_TPS_M wird an eine Maschine gesendet
-100	ERR_WRONG_MACHINE RPC-Returnwert, wenn Maschinenname falsch ist.
-110	ERR_WRONG_HOST RPC-Returnwert, wenn Hostname falsch ist.
-200	ERR_FUNCT_BUSY RPC-Returnwert, wenn gleiche Fkt. Schon läuft mit R_REPORT_H, wenn C_TPORDER_M kommt, bevor der vorherige TPA abgeschlossen ist.
-203	ERR_R_NC4WPC_M R_REPORT_H bei Fehler nach R_NC4WPC_M
-250	ERR_DEL_FILE R_REPORT_H, wenn das File im DH-Server nicht gelöscht werden kann
-262	ERR_NCPROGRAMM_LOAD R_REPORT_H wenn das File im DH nicht geladen werden kann
-263	ERR_NCPROGRAMM_LOAD_SELECT R_REPORT_H wenn das File im DH nicht geladen und angewählt werden kann
-264	ERR_NCPROGRAMM_UNLOAD R_REPORT_H wenn das File im DH nicht entladen werden kann
-265	ERR_NCPROGRAMM_SELECT R_REPORT_H wenn das File im DH nicht angewählt werden kann
-266	ERR_NCPROGRAMM_DESELECT R_REPORT_H wenn das File im DH nicht abgewählt werden kann
-270	ERR_C_ORDER_M_TIME R_REPORT_H wenn Uhrzeit/Datum nicht aktualisiert werden kann
-300	ERR_GET_FILE RPC-Returnwert, wenn das Files nicht vom FLR geholt werden kann.
-301	ERR_NC_RENAME R_REPORT_H beim Kürzen eines langen Dateinamens auf 8.3
-302	ERR_NC_FILESTATUS R_REPORT_H beim Setzen des Dateidatums bei R_DATA_M (SFkt=1)
-310	ERR_DH_CREATE R_REPORT_H bei Fehler in dh_create
-320	ERR_DH_GETFILE R_REPORT_H bei Fehler, wenn Programm nicht in Datenhaltung
-400	ERR_PUT_FILE RPC-Returnwert, wenn das Files nicht auf den FLR gebracht werden kann
-500	ERR_DDE_CONNECT R_REPORT_H bei DDE-Connect Fehler für R_DDEDATA_M ()
-510	ERR_DDE_POKE R_REPORT_H bei DDE-Poke Fehler für R_DDEDATA_M ()
-600	ERR_TOOLDATADESCR R_REPORT_H bei WZ-Abfrage mit falscher Datenstrukturnr.
-610	ERR_GET_TOOLDATA R_REPORT_H bei WZ-Abfrage, Lesefehler bei WZ-Daten

Fehler-nummer	Bedeutung
-700	ERR_TPS R_REPORT_H, wenn TPS einen TPA mit Fehler quittiert.
-800	ERR_T_VAR_M R_REPORT_H bei Fehler bei T_VAR_M
-805	ERR_R_VAR_M R_REPORT_H bei Fehler bei R_VAR_M
-810	ERR_VARSET R_REPORT_H bei unbekanntem Variablen Set (SCVARSET.INI)
-820	ERR_VARSET R_REPORT_H bei Fehler im Variablen Set
-6003	ERR_FILE_NOT_FOUND 1. R_REPORT_H nach T_DATA_M (SFkt=10), wahrscheinlich Name1 falsch
-6020	ERR_TOOL_NOT_FOUND R_REPORT_M nach T_DATA_H (SFkt =21 ... 23), wenn das angeforderte Werkzeug nicht gefunden wurde.

A.5 Literatur

Allgemeine Dokumentation

/BU/ SINUMERIK & SIMODRIVE, Automatisierungssysteme für Bearbeitungsmaschinen
Katalog NC 60
Bestellnummer: E86060-K4460-A101-A9
Bestellnummer: E86060-K4460-A101-A9-7600 (englisch)

/IKPI/ Industrielle Kommunikation und Feldgeräte
Katalog IK PI
Bestellnummer: E86060-K6710-A101-B2
Bestellnummer: E86060-K6710-A101-B2-7600 (englisch)

/ST7/ SIMATIC
Produkte für Totally Integrated Automation und Micro Automation
Katalog ST 70
Bestellnummer: E86060-K4670-A111-A8
Bestellnummer: E86060-K4670-A111-A8-7600 (englisch)

/Z/ MOTION-CONNECT
Verbindungstechnik & Systemkomponenten für SIMATIC, SINUMERIK, MASTERDRIVES und SIMOTION
Katalog NC Z
Bestellnummer: E86060-K4490-A001-B1
Bestellnummer: E86060-K4490-A001-B1-7600 (englisch)

Safety Integrated
Applikationshandbuch
Das Sicherheitsprogramm für die Industrien der Welt
Bestellnummer: 6ZB5000-0AA01-0BA0
Bestellnummer: 6ZB5000-0AA02-0BA0 (englisch)

Elektronische Dokumentation

/CD1/ Das SINUMERIK-System (Ausgabe 03. 04)
DOC ON CD
(mit allen SINUMERIK 840D/840Di/810D/802- und SIMODRIVE- Schriften)
Bestellnummer: 6FC5298-7CA00-0AG0

Anwender-Dokumentation

/AUK/ SINUMERIK 840D/810D (Ausgabe 09.99)
Kurzanleitung **Bedienung AutoTurn**
Bestellnummer: 6FC5298-4AA30-0AP2

/AUP/ SINUMERIK 840D/810D (Ausgabe 02.02)
Bedienungsanleitung **Grafisches Programmiersystem AutoTurn**
Programmieren / Einrichten
Bestellnummer: 6FC5298-4AA40-0AP3

/BA/ SINUMERIK 840D/810D (Ausgabe 10.00)
Bedienungsanleitung **MMC**
Bestellnummer: 6FC5298-6AA00-0AP0

/BAD/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Bedienungsanleitung **HMI Advanced**
Bestellnummer: 6FC5298-6AF00-0AP3

/BAH/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Bedienungsanleitung **HT 6**
Bestellnummer: 6FC5298-0AD60-0AP3

/BAK/ SINUMERIK 840D/840Di/810D (Ausgabe 02.01)
Kurzanleitung Bedienung
Bestellnummer: 6FC5298-6AA10-0AP0

/BAM/ SINUMERIK 810D/840D (Ausgabe 08.02)
Bedienen/Programmieren **ManualTurn**
Bestellnummer: 6FC5298-6AD00-0AP0

/BAS/ SINUMERIK 840D/840Di/810D (Ausgabe 11.03)
Bedienen/Programmieren **ShopMill**
Bestellnummer: 6FC5298-6AD10-0AP2

/BAT/ SINUMERIK 840D/810D (Ausgabe 06.03)
Bedienen/Programmieren **ShopTurn**
Bestellnummer: 6FC5298-6AD50-0AP2

/BEM/ SINUMERIK 840D/810D (Ausgabe 03.04)
Bedienungsanleitung **HMI Embedded**
Bestellnummer: 6FC5298-6AC00-0AP3

/BNM/ SINUMERIK 840D840Di//810D (Ausgabe 03.04)
Benutzeranleitung **Messzyklen**
Bestellnummer: 6FC5298-6AA70-0AP3

/BTDI/ SINUMERIK 840D840Di//810D (Ausgabe 04.03)
Motion Control Information System (MCIS)
Benutzerhandbuch **Tool Data Information**
Bestellnummer: 6FC5297-6AE01-0AP0

/CAD/ SINUMERIK 840D/840Di/810D (Ausgabe 03.02)
Bedienungsanleitung **CAD-Reader**
Bestellnummer: (ist Bestandteil der Online-Hilfe)

/DA/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Diagnoseanleitung
Bestellnummer: 6FC5298-7AA20-0AP0

/KAM/ SINUMERIK 840D/810D (Ausgabe 04.01)
Kurzanleitung **ManualTurn**
Bestellnummer: 6FC5298-5AD40-0AP0

/KAS/ SINUMERIK 840D/810D (Ausgabe 04.01)
Kurzanleitung **ShopMill**
Bestellnummer: 6FC5298-5AD30-0AP0

/KAT/ SINUMERIK 840D/810D (Ausgabe 07.01)
Kurzanleitung **ShopTurn**
Bestellnummer: 6FC5298-6AF20-0AP0

/PG/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Programmieranleitung **Grundlagen**
Bestellnummer: 6FC5298-7AB00-0AP0

/PGA/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Programmieranleitung **Arbeitsvorbereitung**
Bestellnummer: 6FC5298-7AB10-0AP0

/PGK/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Kurzanleitung **Programmierung**
Bestellnummer: 6FC5298-7AB30-0AP0

/PGM/ SINUMERIK 840D/840Di/810D (11.02 Edition)
Programming Guide **ISO Milling**
Bestellnummer: 6FC5298-6AC20-0BP2

/PGT/ SINUMERIK 840D/840Di/810D (11.02 Edition)
Programming Guide **ISO Turning**
Bestellnummer: 6FC5298-6AC10-0BP2

/PGZ/ SINUMERIK 840D840Di//810D (Ausgabe 03.04)
Programmieranleitung **Zyklen**
Bestellnummer: 6FC5298-7AB40-0AP0

/PI/ PCIN 4.4
Software zur Datenübertragung an/von **MMC-Modul**
Bestellnummer: 6FX2060-4AA00-4XB0 (dt., engl., frz.)
Bestellort: WK Fürth

/SYI/ SINUMERIK 840Di (Ausgabe 02.01)
Systemüberblick
Bestellnummer: 6FC5298-6AE40-0AP0

Hersteller-/Service-Dokumentation

a) Listen

/LIS/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
SIMODRIVE 611D
Listen
Bestellnummer: 6FC5297-7AB70-0AP0

b) Hardware

/ASAL/ SIMODRIVE (Ausgabe 10.03)
Projektierungsanleitung Allgemeiner Teil für **Asynchronmotoren**
Bestellnummer: 6SN1197-0AC62-0AP0

/APH2/ SIMODRIVE (Ausgabe 10.03)
Projektierungsanleitung **Asynchronmotoren 1PH2**
Bestellnummer: 6SN1197-0AC63-0AP0

/APH4/ SIMODRIVE (Ausgabe 10.03)
Projektierungsanleitung **Asynchronmotoren 1PH4**
Bestellnummer: 6SN1197-0AC64-0AP0

/APH7M/ MASTERDRIVES VC/MC (Ausgabe 04.04)
Projektierungsanleitung **Asynchronmotoren 1PH7**
Bestellnummer: 6SN1197-0AC66-0AP0

/APH7S/ SIMODRIVE (Ausgabe 01.04)
Projektierungsanleitung **Asynchronmotoren 1PH7**
Bestellnummer: 6SN1197-0AC65-0AP0

/APL6/ MASTERDRIVES VC/MC (Ausgabe 01.04)
Projektierungsanleitung **Asynchronmotoren 1PL6**
Bestellnummer: 6SN1197-0AC67-0AP0

/BH/ SINUMERIK 840D840Di//810D (Ausgabe 11.03)
Handbuch **Bedienkomponenten**
Bestellnummer: 6FC5297-6AA50-0AP3

/BHA/ SIMODRIVE Sensor (Ausgabe 03.03)
Benutzerhandbuch (HW) **Absolutwertgeber mit Profibus-DP**
Bestellnummer: 6SN1197-0AB10-0YP2

/EMV/ SINUMERIK, SIROTEC, SIMODRIVE (Ausgabe 06.99)
Projektierungsanleitung **EMV-Aufbaurichtlinie**
Bestellnummer: 6FC5297-0AD30-0AP1

Die aktuelle Konformitätserklärung finden Sie im Internet unter
http://www4.ad.siemens.de

Bitte geben Sie dort die ID-Nr.: 15257461 in das Feld "Suche" ein (rechts oben) und klicken Sie auf "go".

/GHA/ SINUMERIK/ SIMOTION (Ausgabe 02.03)
ADI4 - Analoge Antriebsschnittstelle für 4 Achsen
Gerätehandbuch
Bestellnummer: 6FC5297-0BA01-0AP1

/PFK6/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (Ausgabe 05.03)
Projektierungsanleitung **Synchron-Servomotoren 1FK6**
Bestellnummer: 6SN1197-0AD05-0AP0

/PFK7/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (Ausgabe 01.03)
Projektierungsanleitung **Synchron-Servomotoren 1FK7**
Bestellnummer: 6SN1197-0AD06-0AP0

/PFS6/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (Ausgabe 04.04)
Projektierungsanleitung **Synchron- Servomotoren 1FS6**
Bestellnummer: 6SN1197-0AD08-0AP1

/PFT5/ SIMODRIVE (Ausgabe 05.03)
Projektierungsanleitung **Synchron-Servomotoren 1FT5**
Bestellnummer: 6SN1197-0AD01-0AP0

/PFT6/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (Ausgabe 01.04)
Projektierungsanleitung **Synchron-Servomotoren 1FT6**
Bestellnummer: 6SN1197-0AD02-0AP0

/PFU/ SINAMICS, SIMOVERT MASTERDRIVES, (Ausgabe 09.03)
MICROMASTER
SIEMOSYN-Motoren 1FU8
Bestellnummer: 6SN1197-0AC80-0AP0

/PHC/ SINUMERIK 810D (Ausgabe 11.02)
Handbuch **Projektierung (HW)**
Bestellnummer: 6FC5297-6AD10-0AP1

/PHD/ SINUMERIK 840D (Ausgabe 11.03)
Handbuch **Projektierung (HW)**
Bestellnummer: 6FC5297-6AC10-0AP3

/PJAL/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (Ausgabe 01.04)
Projektierungsanleitung **Synchron-Servomotoren**
Allgemeiner Teil für 1FT- / 1FK-Motoren
Bestellnummer: 6SN1197-0AD07-0AP0

/PJAS/ SIMODRIVE, MASTERDRIVES (Ausgabe 06.04)
Projektierungsanleitung für Asynchronmotoren
Inhalt: Allgemeiner Teil, 1PH2, 1PH4, 1PH7, 1PL6
Bestellnummer: 6SN1197-0AC61-0AP0

/PJFE/ SIMODRIVE (Ausgabe 03.04)
Projektierungsanleitung **Synchron-Einbaumotoren 1FE1**
Drehstrommotoren für Hauptspindelantriebe
Bestellnummer: 6SN1197-0AC00-0AP5

/PJF1/ SIMODRIVE (Ausgabe 12.02)
Montageanleitung **Synchron-Einbaumotoren 1FE1 051.-1FE1 147**.
Drehstrommotoren für Hauptspindelantriebe
Bestellnummer: 610.43000.02

/PJLM/ SIMODRIVE (Ausgabe 06.02)
Projektierungsanleitung **Linearmotoren 1FN1, 1FN3**
ALL Allgemeines zum Linearmotor
1FN1 Drehstrom Linearmotor 1FN1
1FN3 Drehstrom Linearmotor 1FN3
CON Anschlußtechnik
Bestellnummer: 6SN1197-0AB70-0AP3

/PJM/ SIMODRIVE 611, SIMOVERT MASTERDRIVES (Ausgabe 11.00)
Projektierungsanleitung **Servomotoren**
Drehstrommotoren für Vorschub- und Hauptspindelantriebe
Bestellnummer: 6SN1197-0AA20-0AP4

/PJM2/ SIMODRIVE, MASTERDRIVES MC (Ausgabe 06.04)
Projektierungsanleitung für Synchron-Servomotoren
Inhalt: Allgemeiner Teil, 1FT5, 1FT6, 1FK6, 1FK7, 1FS6
Bestellnummer: 6SN1197-0AC20-0AP0

/PJTM/ SIMODRIVE (Ausgabe 05.03)
Projektierungsanleitung **Einbau-Torquemotoren 1FW6**
Bestellnummer: 6SN1197-0AD00-0AP1

/PJU/ SIMODRIVE 611 (Ausgabe 02.03)
Projektierungsanleitung **Umrichter**
Bestellnummer: 6SN1197-0AA00-0AP6

/PKTM/ SIMODRIVE, SIMOVERT MASTERDRIVES (Ausgabe 03.04)
Projektierungsanleitung **Komplett-Torquemotoren 1FW3**
Bestellnummer: 6SN1197-0AC70-0AP0

/PMH/ SIMODRIVE Sensor (Ausgabe 07.02)
Projektierungs-/Montageanleitung
Hohlwellenmesssystem SIMAG H
Bestellnummer: 6SN1197-0AB30-0AP1

/PMH2/ SIMODRIVE Sensor (Ausgabe 03.04)
Projektierungs-/Montageanleitung
Hohlwellenmesssystem SIMAG H2
Bestellnummer: 6SN1197-0AB31-0AP1

/PMHS/ SIMODRIVE (Ausgabe 12.00)
Montageanleitung **Messsystem für Hauptspindelantriebe**
Zahnradgeber SIZAG2
Bestellnummer: 6SN1197-0AB00-0YP3

/PMS/ SIMODRIVE (Ausgabe 03.04)
Projektierungsanleitung
ECO-Motorspindel für Hauptspindelantriebe
Bestellnummer: 6SN1197-0AD04-0AP1

/PPH/ SIMODRIVE (Ausgabe 12.01)
Projektierungsanleitung **1PH2-/1PH4-/1PH7-Motoren**
Drehstrom-Asynchronmotoren für Hauptspindelantriebe
Bestellnummer: 6SN1197-0AC60-0AP0

/PPM/ SIMODRIVE (Ausgabe 11.01)
Projektierungsanleitung
Hohlwellenmotoren für Hauptspindelantriebe **1PM4 und 1PM6**
Bestellnummer: 6SN1197-0AD03-0AP0

c) Software

/FB1/ SINUMERIK 840D/840Di/810D/FM-NC (Ausgabe 03.04)
Funktionsbeschreibung **Grundmaschine (Teil 1)**
(im folgenden sind die enthaltenen Bücher aufgeführt)
Bestellnummer: 6FC5297-7AC20-0AP0

A2	Diverse Nahtstellensignale
A3	Achsüberwachungen, Schutzbereiche
B1	Bahnsteuerbetrieb, Genauhalt und Look Ahead
B2	Beschleunigung
D1	Diagnosehilfsmittel
D2	Dialogprogrammierung
F1	Fahren auf Festanschlag
G2	Geschwindigkeiten, Soll-/Istwertsysteme, Regelung
H2	Hilfsfunktionsausgabe an PLC
K1	BAG, Kanal, Programmbetrieb
K2	Achsen, Koordinatensysteme, Frames, Werkstücknahes Istwertsystem, Externe Nullpunktverschiebung
K4	Kommunikation
N2	NOT AUS
P1	Planachsen
P3	PLC-Grundprogramm
R1	Referenzpunktfahren
S1	Spindeln
V1	Vorschübe
W1	Werkzeugkorrektur

/FB2/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Funktionsbeschreibung **Erweiterungsfunktionen (Teil 2)**
einschließlich FM-NC: Drehen, Schrittmotor
(im folgenden sind die enthaltenen Bücher aufgeführt)
Bestellnummer: 6FC5297-7AC30-0AP0

A4	Digitale und analoge NCK-Peripherie
B3	Mehrere Bedientafeln und NCUs
B4	Bedienung über PG/PC
F3	Ferndiagnose
H1	Handfahren und Handradfahren
K3	Kompensationen
K5	BAGs, Kanäle, Achstausch
L1	FM-NC lokaler Bus
M1	Kinematische Transformation
M5	Messen
N3	Softwarenocken, Wegschaltsignale
N4	Stanzen und Nibbeln
P2	Positionierachsen
P5	Pendeln
R2	Rundachsen
S3	Synchronspindel
S5	Synchronaktionen (bis SW 3 / danach /FBSY/)
S6	Schrittmotorsteuerung
S7	Speicherkonfiguration
T1	Teilungsachsen
W3	Werkzeugwechsel
W4	Schleifen

/FB3/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Funktionsbeschreibung **Sonderfunktionen (Teil 3)**
(im folgenden sind die enthaltenen Bücher aufgeführt)
Bestellnummer: 6FC5297-7AC80-0AP0
F2 3- bis 5-Achs-Transformation
G1 Gantry-Achsen
G3 Taktzeiten
K6 Konturtunnelüberwachung
M3 Achskopplung und ESR
S8 Konstante Werkstückdrehzahl für Centerless Schleifen
S9 Sollwertschaltung (S9)
T3 Tangentialsteuerung
TE0 Installation und Aktivierung der Compilezyklen
TE1 Abstandsregelung
TE2 Analoge Achse
TE3 Drehzahl-/ Drehmomentkopplung Master-Slave
TE4 Transformationspacket Handling
TE5 Sollwertumschaltung
TE6 MKS-Kopplung
TE7 Wiederaufsetzen - Retrace Support
TE8 Taktunabhängige bahnsynchrone Schaltsignalausgabe
V2 Vorverarbeitung
W5 3D-Werkzeugradiuskorrektur

/FBA/ SIMODRIVE 611D/SINUMERIK 840D/810D (Ausgabe 03.04)
Funktionsbeschreibung **Antriebsfunktionen**
(im folgenden sind die enthaltenen Kapitel aufgeführt)
Bestellnummer: 6SN1197-0AA80-1AP1
DB1 Betriebsmeldungen/Alarmreaktionen
DD1 Diagnosefunktionen
DD2 Drehzahlregelkreis
DE1 Erweiterte Antriebsfunktionen
DF1 Freigaben
DG1 Geberparametrierung
DL1 MD des Linearmotors
DM1 Motor-/Leistungsteilparameter und Reglerdaten berechnen
DS1 Stromregelkreis
DÜ1 Überwachungen/Begrenzungen

/FBAN/ SINUMERIK 840D/SIMODRIVE 611 DIGITAL (Ausgabe 02.00)
Funktionsbeschreibung **ANA-MODUL**
Bestellnummer: 6SN1197-0AB80-0AP0

/FBD/ SINUMERIK 840D (Ausgabe 07.99)
Funktionsbeschreibung **Digitalisieren**
Bestellnummer: 6FC5297-4AC50-0AP0
DI1 Inbetriebnahme
DI2 Scan mit taktilem Sensor (scancad scan)
DI3 Scan mit Laser (scancad laser)
DI4 Fräsprogrammerstellung (scancad mill)

/FBDM/ SINUMERIK 840D/840Di/810D (Ausgabe 09.03)
Funktionsbeschreibung **NC Programmmanagement**
DNC Maschinen
Bestellnummer: 6FC5297-1AE81-0AP0

/FBDN/ SINUMERIK 840D/840Di/810D (Ausgabe 03.03)
Motion Control Information System (MCIS)
Funktionsbeschreibung **NC-Programmmagagement DNC**
Bestellnummer: 6FC5297-1AE80-0AP0
DN1 DNC Plant / DNC Cell
DN2 DNC IFC SINUMERIK, NC-Datenübertragung über Netzwerk

/FBFA/ SINUMERIK 840D/840Di/810D (Ausgabe 11.02)
Funktionsbeschreibung **ISO-Dialekte für SINUMERIK**
Bestellnummer: 6FC5297-6AE10-0AP3

/FBFE/ SINUMERIK 840D/810D (Ausgabe 04.03)
Motion Control Information System (MCIS)
Funktionsbeschreibung **Ferndiagnose**
Bestellnummer: 6FC5297-0AF00-0AP2
FE1 Ferndiagnose (ReachOut)
FE3 RCS Host / RCS Viewer (pcAnywhere)

/FBH/ SINUMERIK 840D/840Di/810D
HMI-Programmierpaket (Ausgabe 11.02)
Bestellnummer: (ist Bestandteil der SW-Lieferung)

Teil 1 Benutzeranleitung
Teil 2 Funktionsbeschreibung

/FBH1/ SINUMERIK 840D/840Di/810D
HMI-Programmierpaket (Ausgabe 03.03)
ProTool/Pro Option SINUMERIK
Bestellnummer: (ist Bestandteil der SW-Lieferung)

/FBHL/ SINUMERIK 840D/SIMODRIVE 611 digital (Ausgabe 10.03)
Funktionsbeschreibung **HLA-Modul**
Bestellnummer: 6SN1197-0AB60-0AP3

/FBIC/ SINUMERIK 840D/840Di/810D (Ausgabe 06.03)
Motion Control Information System (MCIS)
Funktionsbeschreibung **TDI Ident Connection**
Bestellnummer: 6FC5297-1AE60-0AP0

/FBMA/ SINUMERIK 840D/810D (Ausgabe 08.02)
Funktionsbeschreibung **ManualTurn**
Bestellnummer: 6FC5297-6AD50-0AP0

/FBO/ SINUMERIK 840D/810D (Ausgabe 09.01)
Funktionsbeschreibung Projektierung **Bedienoberfläche OP 030**
(im folgenden sind die enthaltenen Kapitel aufgeführt)
Bestellnummer: 6FC5297-6AC40-0AP0
BA Bedienanleitung
EU Entwicklungsumgebung (Projektierpaket)
PSE Einführung in die Projektierung der Bedienoberfläche
(IK Installationspaket: Softwareupdate und Konfiguration)

/FBP/ SINUMERIK 840D (Ausgabe 03.96)
Funktionsbeschreibung **C-PLC-Programmierung**
Bestellnummer: 6FC5297-3AB60-0AP0

/FBR/ SINUMERIK 840D/840Di/810D (Ausgabe 01.04)
Funktionsbeschreibung **Rechnerkopplung RPC SINUMERIK**
Bestellnummer: 6FC5297-6AD61-0AP0
NFL Nahtstelle zum Fertigungsleitrechner
NPL Nahtstelle zu PLC/NCK

/FBSI/ SINUMERIK 840D/SIMODRIVE (Ausgabe 11.03)
Funktionsbeschreibung SINUMERIK **Safety Integrated**
Bestellnummer: 6FC5297-6AB80-0AP2

/FBSP SINUMERIK 840D/840Di/810D (Ausgabe 08.03)
Funktionsbeschreibung **ShopMill**
Bestellnummer: 6FC5297-6AD80-0AP1

/FBST/ SIMATIC (Ausgabe 01.01)
Funktionsbeschreibung **FM STEPDRIVE/SIMOSTEP**
Bestellnummer: 6SN1197-0AA70-0YP4

/FBSY/ SINUMERIK 840D/810D (Ausgabe 03.04)
Funktionsbeschreibung **Synchronaktionen**
Bestellnummer: 6FC5297-7AD40-0AP2

/FBT/ SINUMERIK 840D/810D (Ausgabe 01.02)
Funktionsbeschreibung **ShopTurn**
Bestellnummer: 6FC5297-6AD70-0AP1

/FBTC/ SINUMERIK 840D/810D (Ausgabe 01.02)
IT-Solutions
Funktionsbeschreibung **Tool Data Communication SinTDC**
Bestellnummer: 6FC5297-5AF30-0AP0

/FBTD/ SINUMERIK 840D/810D (Ausgabe 02.01)
IT-Solutions
Funktionsbeschreibung **Werkzeugbedarfsermittlung (SinTDI)**
mit Online-Hilfe
Bestellnummer: 6FC5297-6AE00-0AP0

/FBTP/ SINUMERIK 840D/840Di/810D (Ausgabe 01.03)
Motion Control Information System (MCIS)
Funktionsbeschreibung **Vorbeugende Instandhaltung TPM**
Bestellnummer: Dokument ist Bestandteil der Software

/FBU/ SIMODRIVE 611 universal/universal E (Ausgabe 07.03)
Funktionsbeschreibung **Regelungskomponente für Drehzahlregelung und Positionieren**
Bestellnummer: 6SN1197-0AB20-0AP8

/FBU2/ SIMODRIVE 611 **universal** (Ausgabe 04.02)
Montageanleitung
(liegt jedem SIMODRIVE 611 universal bei)

/FBW/ SINUMERIK 840D/810D (Ausgabe 11.02)
Funktionsbeschreibung **Werkzeugverwaltung**
Bestellnummer: 6FC5297-6AC60-0AP1

/HBA/ SINUMERIK 840D/840Di/810D (Ausgabe 03.02)
Handbuch **@Event**
Bestellnummer: 6AU1900-0CL20-0AA0

/HBI/ SINUMERIK 840Di (Ausgabe 09.03)
Handbuch **SINUMERIK 840Di**
Bestellnummer: 6FC5297-6AE60-0AP2

/INC/ SINUMERIK 840D840Di//810D (Ausgabe 06.03)
Systembeschreibung **Inbetriebnahme-Tool SINUMERIK SinuCOM NC**
Bestellnummer: (Bestandteil der Online-Hilfe des IBN-Tools)

/PGA1/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Listenhandbuch **Systemvariablen**
Bestellnummer: 6FC5297-7AE10-0AP0

/PJE/ SINUMERIK 840D/810D (Ausgabe 08.01)
Funktionsbeschreibung **Projektierpaket HMI Embedded**
Softwareupdate, Konfiguration, Installation
Bestellnummer: 6FC5297-6EA10-0AP0

/PS/ SINUMERIK 840D/810D (Ausgabe 09.03)
Projektierungsanleitung Projektiersyntax
Diese Schrift ist Bestandteil der SW-Lieferung und als PDF verfügbar

/POS1/ SIMODRIVE **POSMO A** (Ausgabe 08.03)
Benutzerhandbuch **Dezentraler Positioniermotor am PROFIBUS DP**
Bestellnummer: 6SN2197-0AA00-0AP6

/POS2/ SIMODRIVE **POSMO A** (Ausgabe 05.03)
Montageanleitung (liegt jedem POSMO A bei)

/POS3/ SIMODRIVE POSMO SI/CD/CA (Ausgabe 07.03)
Benutzerhandbuch **Dezentrale Servo Antriebstechnik**
Bestellnummer: 6SN2197-0AA20-0AP5

/POS4/ SIMODRIVE **POSMO SI** (Ausgabe 04.02)
Montageanleitung (liegt jedem POSMO SI bei)

/POS5/ SIMODRIVE **POSMO CD/CA** (Ausgabe 04.02)
Montageanleitung (liegt jedem POSMO CD/CAbei)

/S7H/ SIMATIC S7-300 (Ausgabe 2002)
Installationshandbuch **Technologische Funktionen**
Bestellnummer: 6ES7398-8AA03-8AA0
- Referenzhandbuch: CPU-Daten (HW-Beschreibung)
- Referenzhandbuch: Baugruppendaten

/S7HT/ SIMATIC S7-300 (Ausgabe 03.97)
Handbuch **STEP 7, Grundwissen, V. 3.1**
Bestellnummer: 6ES7810-4CA02-8AA0

/S7HR/ SIMATIC S7-300 (Ausgabe 03.97)
Handbuch **STEP 7, Referenzhandbücher, V. 3.1**
Bestellnummer: 6ES7810-4CA02-8AR0

/S7S/ SIMATIC S7-300 (Ausgabe 04.02)
Positionierbaugruppe **FM 353 für Schrittantrieb**
Bestellung zusammen mit dem Projektierpaket

/S7L/ SIMATIC S7-300 (Ausgabe 04.02)
Positionierbaugruppe **FM 354 für Servoantrieb**
Bestellung zusammen mit dem Projektierpaket

/S7M/ SIMATIC S7-300 (Ausgabe 01.03)
Mehrfachbaugruppe FM 357.2 für Servo- bzw. Schrittantrieb
Bestellung zusammen mit dem Projektierpaket

/SP/ SIMODRIVE 611-A/611-D
SimoPro 3.1
Programm zur Projektierung von Werkzeugmaschinen-Antrieben
Bestellnummer: 6SC6111-6PC00-0AA❑
Bestellort: WK Fürth

d) Inbetrieb-nahme

/BS/ SIMODRIVE 611 analog (Ausgabe 10.00)
Beschreibung **Inbetriebnahmesoftware für Hauptspindel- und Asynchronmotormodule Version 3.20**
Bestellnummer: 6SN1197-0AA30-0AP1

/IAA/ SIMODRIVE 611A (Ausgabe 10.00)
Inbetriebnahmeanleitung
Bestellnummer: 6SN1197-0AA60-0AP6

/IAC/ SINUMERIK 810D (Ausgabe 11.02)
Inbetriebnahmeanleitung
(einschl. Beschreibung der Inbetriebnahme-Software SIMODRIVE 611D)
Bestellnummer: 6FC5297-6AD20-0AP1

/IAD/ SINUMERIK 840D/SIMODRIVE 611D (Ausgabe 03.04)
I**nbetriebnahmeanleitung**
(einschl. Beschreibung der Inbetriebnahme-Software SIMODRIVE 611D)
Bestellnummer: 6FC5297-7AB10-0AP0

/IAM/ SINUMERIK 840D/840Di/810D (Ausgabe 03.04)
Inbetriebnahmeanleitung **HMI**
Bestellnummer: 6FC5297-6AE20-0AP3
AE1 Aktualisierungen/Ergänzungen
BE1 Bedienoberfläche ergänzen
HE1 Online-Hilfe
IM2 Inbetriebnahme HMI Embedded
IM4 Inbetriebnahme HMI Advanced
TX1 Fremdsprachentexte erstellen

■

I

I Index

I.1 Stichwortindex

A

B

C

D

F

G

H

I

M

N

P

■

An
Siemens AG

A&D MC BMS
Postfach 3180

D-91050 Erlangen

Tel. +49 (0) 180 / 5050 – 222 [Hotline]

Fax +49 (0) 9131 / 98 – 2176 [Dokumentation]
E-Mail: motioncontrol.docu@erlf.siemens.de

Vorschläge

Korrekturen

für Druckschrift:

Motion Control Information System

SINUMERIK 840D/840Di/810D
Rechnerkopplung RPC SINUMERIK

Hersteller-Dokumentation

Funktionsbeschreibung

Bestell-Nr.: 6FC5297-6AD61-0AP0
Ausgabe 01.04

Absender

Name:

Anschrift Ihrer Firma/Dienststelle

Straße: ______________________________

PLZ: ___________ Ort: ______________________

Telefon: _________ / ______________________

Telefax: ________ / ______________________

Sollten Sie beim Lesen dieser Unterlage auf Druckfehler gestoßen sein, bitten wir Sie, uns diese mit diesem Vordruck mitzuteilen. Ebenso dankbar sind wir für Anregungen und Verbesserungen.

Vorschläge und/oder Korrekturen

Dokumentationsübersicht SINUMERIK 840D/840Di/810D (03.2004)

Allgemeine Dokumentation

- *SINUMERIK* 840D/810D — Werbeschrift
- *SINUMERIK* 840D/840Di/810D/ — Katalog Bestellu. NC 60 ***)**
- *SINUMERIK SIROTEC SIMODRIVE* Zubehör — Katalog Zubehör NC–Z

Anwender–Dokumentation

- *SINUMERIK* 840D/810D/FM–NC — **AutoTurn**
  - – Kurzanleitung
  - – Programmieren /Einrichten
- *SINUMERIK* 840D/840Di 810D — Bedienungsanl. **– HT 6**
- *SINUMERIK* 840D/840Di/810D — **Diagnose–anleitung *)**
- *SINUMERIK* 840D/840Di/810D — **Bedienungsanl. *)**
  - – Kurzanleitung
  - – HMI Embedded
  - – HMI Advanced

Anwender–Dokumentation

- *SINUMERIK* 840D/840Di/810D — **Programmieranl.**
  - – Kurzanleitung
  - – Grundlagen ***)**
  - – Arbeitsvorbereit. ***)**
  - – Zyklen
  - – Meßzyklen
  - – ISO Turning/Milling
- *SINUMERIK* 840D/810D — Bedienungsanl.
  - – **ManualTurn**
  - – Kurzanl. ManualTurn
  - – **ShopMill**
  - – Kurzanl. ShopMill
  - – **ShopTurn**
  - – Kurzanl. ShopTurn
- *SINUMERIK* 840Di — **Systemüberblick**

Hersteller–/Service–Dokumentation

- *SINUMERIK* — **Projektierung** (HW) ***)**
  - – 810D
  - – 840D
- *SINUMERIK* 840D/840Di/810D — **Bedien–komponenten** (HW) ***)**
- *SINUMERIK* 840D/810D — Funktionsbeschr.
  - – **ManualTurn**
  - – **ShopMill**
  - – **ShopTurn**
- *SINUMERIK* 840D/840Di/810D — Funktionsbeschr. **Synchronaktionen**

Hersteller–/Service–Dokumentation

- *SINUMERIK SIMODRIVE* 611D 840D/810D — Funktionsbeschr. **Antriebsfunktion *)**
- *SINUMERIK* 840D/840Di/810D — **Funktionsbeschr.**
  - – Grundmaschine ***)**
  - – Erweiterungsfunkt.
  - – Sonderfunktionen
- *SINUMERIK* 840D/840Di/810D — Funktionsbeschr. **Werkzeug–verwaltung**
- *SINUMERIK* 840D/810D — **Projektier–Pkt.** HMI Embedded
- *SINUMERIK* 840D/810D — Funktionsbeschr. Projekt. Bedien–oberfläche **OP 030**
- *SINUMERIK* 840D/810D — **MCIS**
  - – Rechnerkopplung
  - – Werkzeugbedarfse.
  - – NC–Datenverwalt.
  - – NC–Datenübertrag.
  - – Tool Data Communica.

Hersteller–/Service–Dokumentation

- *SINUMERIK SIMODRIVE* — Funktionsbeschr. SINUMERIK **Safety Integrated**
- *SINUMERIK* 840D — Funktionsbeschr. **Digitalisieren**
- *SINUMERIK SIMODRIVE* — **Inbetriebnameanl. *)**
  - – 810D
  - – 840D/611D
  - – HMI
- *SINUMERIK SIMODRIVE* 840D/840Di 810D 611D — **– Listen *)** **– Listen Systemvar.**
- *SINUMERIK SIMODRIVE* 840D 611D — Funktionsbeschr. **Linearmotor**
- *SINUMERIK SIMODRIVE* 840D 611D — Funktionsbeschr.
  - **– Hydraulikmodul**
  - **– Analogmodul**
- *MOTION CONTROL SYSTEMS* — **EMV**–Richtlinien

Elektronische Dokumentation

- *SINUMERIK SIMODRIVE* 840D/840Di/810D 611, Motoren — **DOC ON CD *)** Das SINUMERIK–System

Hersteller–/Service–Dokumentation

- *SINUMERIK* 840D/840Di/810D — Funktionsbeschr. **ISO–Dialekte** für SINUMERIK
- *SINUMERIK* 840Di — **Handbuch** (HW + Inbetriebn.)
- *SINUMERIK* 840D/810D — Funktionsbeschr. **Ferndiagnose**
- *SINUMERIK* 840D/840Di/810D — **Handbuch @ Event**

***)** Empfohlener Minimalumfang der Dokumentation

Siemens AG
Automatisierungs- und Antriebstechnik
Motion Control Systems
Postfach 3180, D – 91050 Erlangen
Bundesrepublik Deutschland

www.ad.siemens.de

Bestell-Nr.: 6FC5297-6AD61-0AP0

Gedruckt in der Bundesrepublik Deutschland